LPDDR3 HPHY

LPDDR3, LPDDR2 PHY (V5R1, V5R2)

Revision 0.33

January 2013

DataSheet

SAMSUNG ELECTRONICS RESERVES THE RIGHT TO CHANGE PRODUCTS, INFORMATION AND SPECIFICATIONS WITHOUT NOTICE.

Products and specifications discussed herein are for reference purposes only. All information discussed herein is provided on an "AS IS" basis, without warranties of any kind.

This document and all information discussed herein remain the sole and exclusive property of Samsung Electronics. No license of any patent, copyright, mask work, trademark or any other intellectual property right is granted by one party to the other party under this document, by implication, estoppel or otherwise.

Samsung products are not intended for use in life support, critical care, medical, safety equipment, or similar applications where product failure could result in loss of life or personal or physical harm, or any military or defense application, or any governmental procurement to which special terms or provisions may apply.

For updates or additional information about Samsung products, contact your nearest Samsung office.

All brand names, trademarks and registered trademarks belong to their respective owners.

Important Notice

Samsung Electronics Co. Ltd. (“Samsung”) reserves the right to make changes to the information in this publication at any time without prior notice. All information provided is for reference purpose only. Samsung assumes no responsibility for possible errors or omissions, or for any consequences resulting from the use of the information contained herein.

This publication on its own does not convey any license, either express or implied, relating to any Samsung and/or third-party products, under the intellectual property rights of Samsung and/or any third parties.

Samsung makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Samsung assume any liability arising out of the application or use of any product or circuit and specifically disclaims any and all liability, including without limitation any consequential or incidental damages.

Customers are responsible for their own products and applications. "Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by the customer's technical experts.

Samsung products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Samsung product could reasonably be expected to create a situation where personal injury or death may occur. Customers acknowledge and agree that they are solely responsible to meet all other legal and regulatory requirements regarding their applications using Samsung products notwithstanding any information provided in this publication. Customer shall indemnify and hold Samsung and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, expenses, and reasonable attorney fees arising out of, either directly or indirectly, any claim (including but not limited to personal injury or death) that may be associated with such unintended, unauthorized and/or illegal use.

WARNING No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electric or mechanical, by photocopying, recording, or otherwise, without the prior written consent of Samsung. This publication is intended for use by designated recipients only. This publication contains confidential information (including trade secrets) of Samsung protected by Competition Law, Trade Secrets Protection Act and other related laws, and therefore may not be, in part or in whole, directly or indirectly publicized, distributed, photocopied or used (including in a posting on the Internet where unspecified access is possible) by any unauthorized third party. Samsung reserves its right to take any and all measures both in equity and law available to it and claim full damages against any party that misappropriates Samsung’s trade secrets and/or confidential information.

警 告 本文件仅向经韩国三星电子株式会社授权的人员提供，其内容含有商业秘密保护相关法规规定并受其保护的三星电子株式会社商业秘密，任何直接或间接非法向第三人披露、传播、复制或允许第三人使用该文件全部或部分内容的行为（包括在互联网等公开媒介刊登该商业秘密而可能导致不特定第三人获取相关信息的行为）皆为法律严格禁止。此等违法行为一经发现，三星电子株式会社有权根据相关法规对其采取法律措施，包括但不限于提出损害赔偿请求。

Samsung Electronics Co., Ltd.
San #24 Nongseo-Dong, Giheung-Gu
Yongin-City, Gyeonggi-Do, Korea 446-711

Contact Us: mail@samsung.com
TEL: (82)-(31)-209-XXXX
FAX: (82)-(31)-209-XXXX

Home Page: http://www.samsungsemi.com

Trademarks

All brand names, trademarks and registered trademarks belong to their respective owners.

- Exynos, Exynos4210, FlexOneNAND, and OneNAND are trademarks of Samsung Electronics.
- ARM, Jazelle, TrustZone, and Thumb are registered trademarks of ARM Limited. Cortex, ETM, ETB, Coresight, ISA, and Neon are trademarks of ARM Limited.
- Java is a trademark of Sun Microsystems, Inc.
- SD is a registered trademark of Toshiba Corporation.
- MMC and eMMC are trademarks of MultiMediaCard Association.
- JTAG is a registered trademark of JTAG Technologies, Inc.
- Synopsys is a registered trademark of Synopsys, Inc.
- I2S is a trademark of Phillips Electronics.
- I2C is a trademark of Phillips Semiconductor Corp.
- MIPI and Slimbus are registered trademarks of the Mobile Industry Processor Interface (MIPI) Alliance.

All other trademarks used in this publication are the property of their respective owners.

Chip Handling Guide

Precaution against Electrostatic Discharge

When handling semiconductor devices, be sure that the environment is protected against static electricity.

1. Operators should wear anti-static clothing and use earth band.
2. All objects that come in direct contact with devices should be made of materials that do not produce static electricity that would cause damage.
3. Equipment and work table must be earthed.
4. Ionizer is recommended to remove electron charge.

Contamination

Be sure to use semiconductor products in the environment that may not be exposed to dust or dirt adhesion.

Temperature/Humidity

Semiconductor devices are sensitive to environment temperature and humidity. High temperature or humidity may deteriorate semiconductor device’s characteristics. Therefore avoid storage or use in such conditions.

Mechanical Shock

Care should be exercised not to apply excessive mechanical shock or force on semiconductor device.

Chemical

Do not expose semiconductor device to chemical because reaction to chemical may cause deterioration of device characteristics.

Light Protection

In case of non-EMC (Epoxy Molding Compound) package, do not expose semiconductor IC to strong light. It may cause device’s malfunction. (But, some special products which utilize the light or have security function are excepted from this guide)

Radioactive, Cosmic and X-ray

Semiconductor devices can be influenced by radioactive, cosmic ray or X-ray. Radioactive, cosmic and X-ray may cause soft error during device operation. Therefore semiconductor devices must be shielded under environment that may be exposed to radioactive, cosmic ray or X-ray.

EMS (Electromagnetic Susceptibility)

Note that semiconductor device’s characteristics may be affected by strong electromagnetic wave or magnetic field during operation under insufficient PCB circuit design for EMS.

Revision History

Revision No.	Date	Description	Author(s)
0.10	2011-01-01	• Preliminary Spec	S.H. Kim
0.11	2011-09-16	• Correct the constraints when ctrl_atgate=1.(p1) • Add the configuration for "byte_rdlvl_en".(p90) • Correct Write leveling for easy explanations.(p91) • Correct CA Calibration for easy explanations.(p92)	S.H. Kim
0.12	2011-09-21	• Update CA Calibration due to changing Memory Spec.(p92)	S.H. Kim
0.13	2011-10-01	• Correct PHY_CON31 ~ PHY_CON34.(p63) • Add explanations about the condition of input clock.(p68) • Add some guides about "DQS Cleaning"(p75) • Add constraints during leveling(p90)	S.H. Kim
0.14	2011-10-18	• Correct Figure 7-3. PHY Read Feedback. (p84) • Correct Figure 7-4. PHY Write Feedback. (p85) • Correct Figure 7-7. I/O Test. (p87)	S.H. Kim
0.15	2011-11-23	• Correct descriptions in "ctrl_atgate"(p41) • Add descriptions about "GATE Cycle Adjust"(p44) • Add Cautions in Read SDLL Code Control Register(p48) • Add Cautions in Write SDLL Code Control Register(p50) • Remove "ctrl_clock and ctrl_flock become HIGH" about ctrl_dll_on(p54) • Add register field in Read Data Enable Timing Status Register(p58) • Add "ca_swap_mode" register(p60) • Add descriptions to read register(p59,59,61) • Add more descriptions about ZQ Control Register(p56) • Correct "zq_done" as Read Only(p58) • Add MPR setting in GATE LEVELING(p94) • Add more descriptions in Write DQ Calibration (p95) • Correct LOW FREQUENCY OPERATION(p96)	S.H. Kim
0.16	2012-01-19	• Remove pull-up/pull-down test during high-z(p39) • Change default value in PHY_CON0[12](p41) • Change default value of "ctrl_ddr_mode"(p41) • Correct information about "ctrl_read_disable"(p41) • Add the duration of GATE when doing Gate Leveling(p77) • Add Memory setting during initialization(p90) • Add ZQ manual setting procedure(p100) • Add default values during feedback test(p84) • Correct information about PDQS(p88) • Correct descriptions about write On-Die Termination(p101) • Add the cautions about read On-Die Termination(p103)	S.H. Kim

0.17		• Correct Figure 8-6(p104)	S.H.Kim
0.20	2012-03-12	• Correct Figure 1-1(p18) • Add descriptions about "clkm"(p25) • Correct descriptions about "dfi_freq_ratio"(p29) • Correct descriptions when reading "PHY_CON34"(p63) • Add ZQ I/O Test(p79) • Correct Test Procedures(p82) • Correct Figures in Feedback Test(p84) • Add "DLLDeskewEn" in Initialization procedures(p90) • Add the compensation range by Write Leveling(p91) • Correct descriptions about Low Frequency Operation(p96) • Add Self-Refresh mode during going back to the original frequency(p96)	S.H.Kim
0.21	2012-03-15	• Add DDR3 support (p18)	J.H.Oh
0.22	2012-03-16	• Add Driver Strength Selection Information(p63)	S.H.Kim
0.23	2012-03-21	• Add the recommended patterns "rdlvl_rddata_adj"(p43) • Add descriptions about "ReadModeCon" for more clarifications(p44) • Add descriptions about ddr3_cmd, lpddr2_cmd and cmd_default(p90) • Correct the recommendation value of "rdlvl_incr_adj"(p94)	S.H.Kim
0.24	2012-03-28	• Correct descriptions about "ReadModeCon" for more clarifications(p44) • Add information about each bit in "ddr3_cmd" and "lpddr3_cmd"(p60) • Add setting "dfi_cke_p0/p1" during CA Calibration(p92) • Add setting "lpddr2_addr" during Calibration for more clarifications(p94)	S.H.Kim
0.25	2012-04-28	• Add Feedback Test guide in DDR3(p84) • Add cautions about "dfi_ctrlupd_req" during training(p90) • Correct setting about " rdlvl_rddata_adj"(p41, p94)	S.H.Kim
0.26	2012-06-27	• Remove "Interface Signals Control According to Operation Mode" Table(p34) • Correct description in "DLLDeskewEn"(p43) • Correct description in "InitDeskewEn"(p43) • Remove "FastDeskewEn" and "FastDeskewStart"(p43) • Remove Feedback Control register(p56) • Correct CA[9:0] to ADCT[15:0] (p63) • Correct Note in Table 7-2(p78) • Add ZQ I/O Test mode(p78) • Add "cautions"(p87, p89)	S.H.Kim
0.27	2012-07-03	• Correct Figure 7-8 (p88)	H.K.Lee
0.28	2012-07-04	• Correct description in "I/O Test mode" (p39)	H.K.Lee
0.29	2012-07-17	• Add Description about "ReadModeCon"(p44)	S.H.Kim

		• Correct ZQ Calibration(p90) • Correct the value of rdlvl_pass_adj(p90) • Correct ZQ Calibration Procedure(p99) • Add the condition of "ctrl_wake_up=0"(p105) • Correct Timing Diagram(p106) • Add the condition of "zq_mode_noterm(p56)	
0.30	2012-08-07	• Add Cautions about reset during Feedback(p82) • Correct typo about dfi_odt_p0/p1(p91) • Remove the setting of ctrl_offset*(p98) • Correct description about ctrl_read_* for easy explanation (p103) • Correct Figure 8-6(p104) • Add PHY_CON26[13:0]) value.(p90)	S.H.Kim
0.31	2012-09-24	• GATEI,GATEO, io_gate_in and io_gate_out pins are removed. Figure 2-2(p21), Figure 7-5(p86)	H.K.Lee
0.32	2012-12-26	• Correct Table 7-1(p78) • Correct default of "test_ext_offsetw", "test_ext_offsetr" and "test_ext_offsetd"(p80) • Correct the available skew between CK and DQS(p91) • Correct descriptions about " test_ext_rdlvl_vwml" and " test_ext_rdlvl_vwmr", please don't assign these pins as test pins(p80)	S.H.Kim
0.33	2013-01-17	• Correct typo about "zq_mode_term", 3'b100 → 3'b001(p56)	S.H.Kim

Revision Descriptions for Revision 1.00

Chapter Name	Page	Major Changes comparing with Last Version
01_Product Overview	1-1	
	1-2	

Table of Contents

List of Figures

List of Tables

List of Conventions

Register RW Access Type Conventions

Type	Definition	Description
R	Read Only	The application has permission to read the Register field. Writes to read-only fields have no effect.
W	Write Only	The application has permission to write in the Register field.
RW	Read & Write	The application has permission to read and writes in the Register field. The application sets this field by writing 1’b1 and clears it by writing 1’b0.

Register Value Conventions

Expression	Description
x	Undefined bit
X	Undefined multiple bits
?	Undefined, but depends on the device or pin status
Device dependent	The value depends on the device
Pin value	The value depends on the pin status

Reset Value Conventions

Expression	Description
0	Clears the register field
1	Sets the register field
x	Don't care condition

Warning: Some bits of control registers are driven by hardware or write operation only. As a result the indicated reset value and the read value after reset might be different.

List of Terms

Terms	Descriptions

List of Acronyms

Acronyms	Descriptions

1 Overview

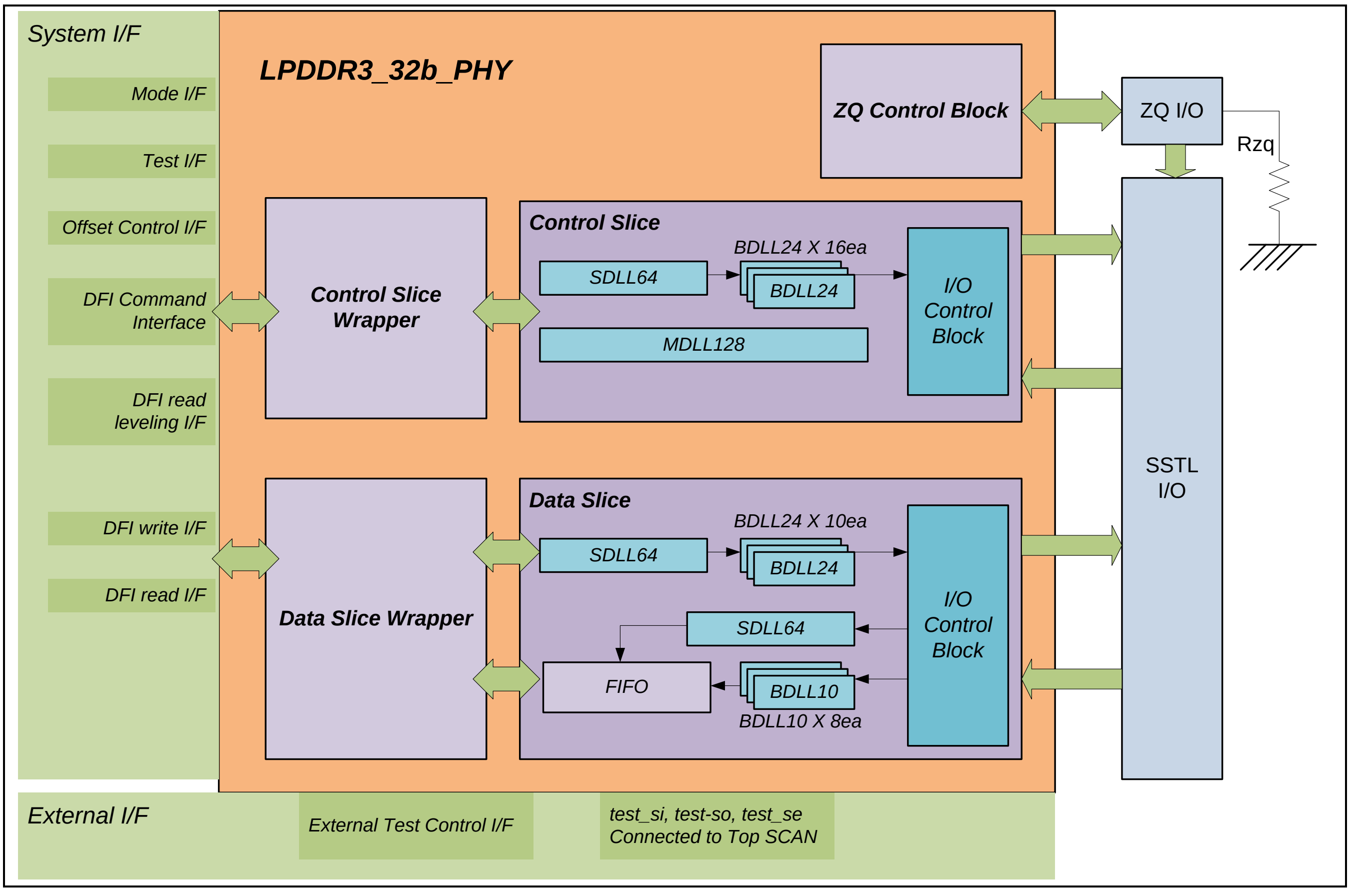

Figure 1-1 LPDDR3 HPHY

The LPDDR3 HPHY(V5R1, V5R2) provides the following features:

- LPDDR2, LPDDR3, DDR3 are supported.
- Fully digital DLL for 90° phase shift of strobe signal.
- FIFO (width:16-bit/depth:16 per 8-bit data slice) for programmable read timing.
- Provide feedback loop-back test scheme for at-speed data and control channel test.

NOTE: When "ctrl_atgate=1", consider the following constraints.
Support the fixed burst length (BL=4, 8, 16).
RL should be greater than 4.
If PHY is used with the dual-rank configurations, RL(Read Latency) and BL(Burst Length) should be used as the same value for those two ranks.

Warning: Single phase 50:50 duty clock(=clk2x) is needed. (at least, should be 49:51 ~ 51:49)

2 BLOCK DIAGRAM DESCRIPTION

2.1 DLL & CONTROL I/F

2.1.1 DLL

DLL detects one clock period and generates delay line control signal. Delay line consists of 128 delay cells and is controlled by control logic. Delay line delays input clock and phase detector compares input clock and delayed clock. After phase comparison, phase detector detects whether delayed clock is lag or lead and generates INC/DEC signals to increase or decrease the delay amount of the delay line.

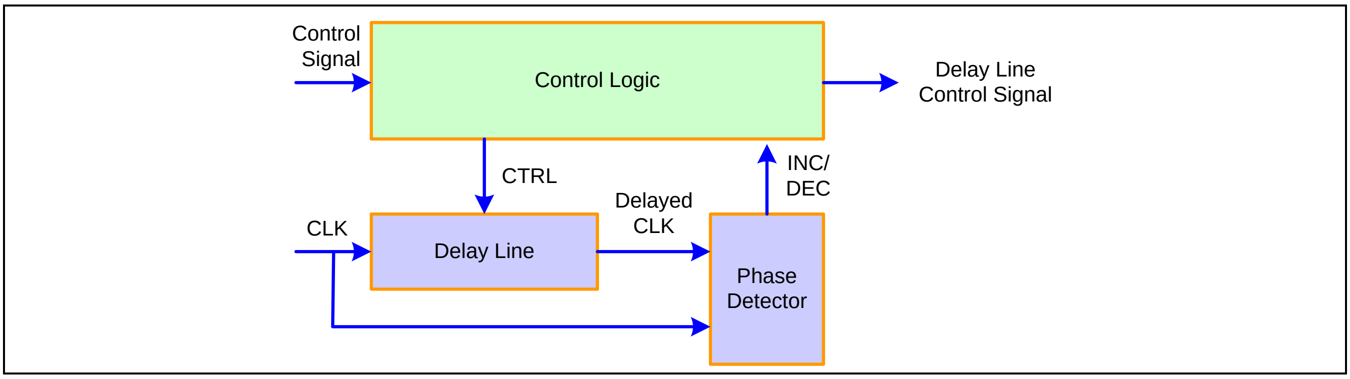

Figure 2-1. Block Diagram of DLL

2.1.2 Control

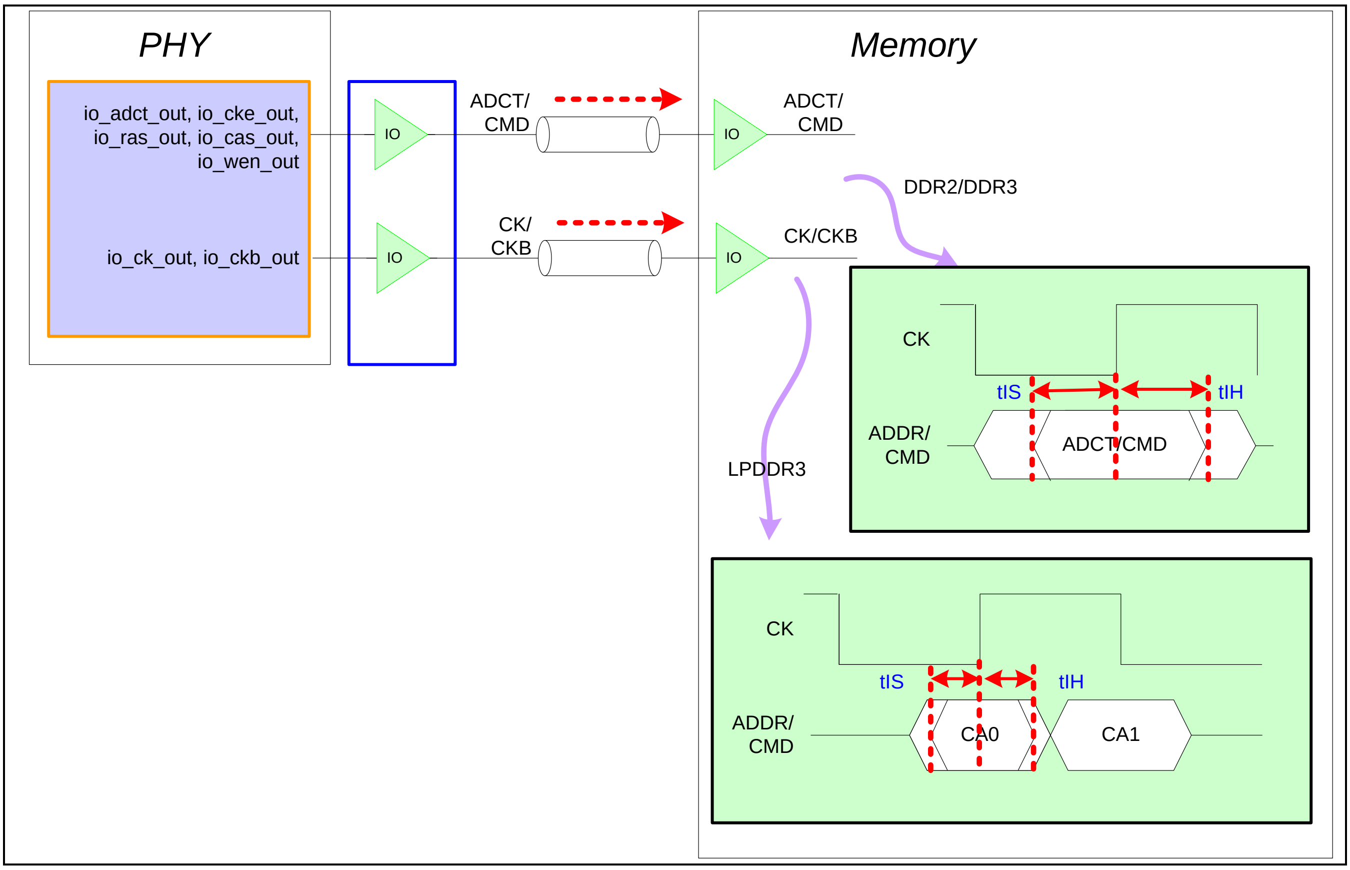

Figure 2-2. Block Diagram of the Control Path

Control block generates address, control and clock signals to I/Os. ADCT[15:0] are reset(RSN)/ODT/address (ADDR)/bank(BA)/chip-select(CS) signals and RAS, CAS, WE are command signals. CK and CKB are differential clock signals. Address and control signals are center aligned at rising CK edge to maximize address/control signal setup/hold timing.

Strobe signal of DDR has high-Z state because it's bi-directional and this high-z state should be cleaned to be used as a clock signal for memory read transaction.

2.2 DATA I/F

2.2.1 Write Path

Data I/F block generates data, mask and strobe signals to interface memory for memory write transaction. DQs are 8-bit width data signals, DM is data mask signal and DQS is strobe signal. Therefore, to interface 32-bit memory, 4 data slices are required.

DQS is edge-aligned signal to CK/CKB and DQs/DM are center-aligned signals to the edge of CK/CKB.

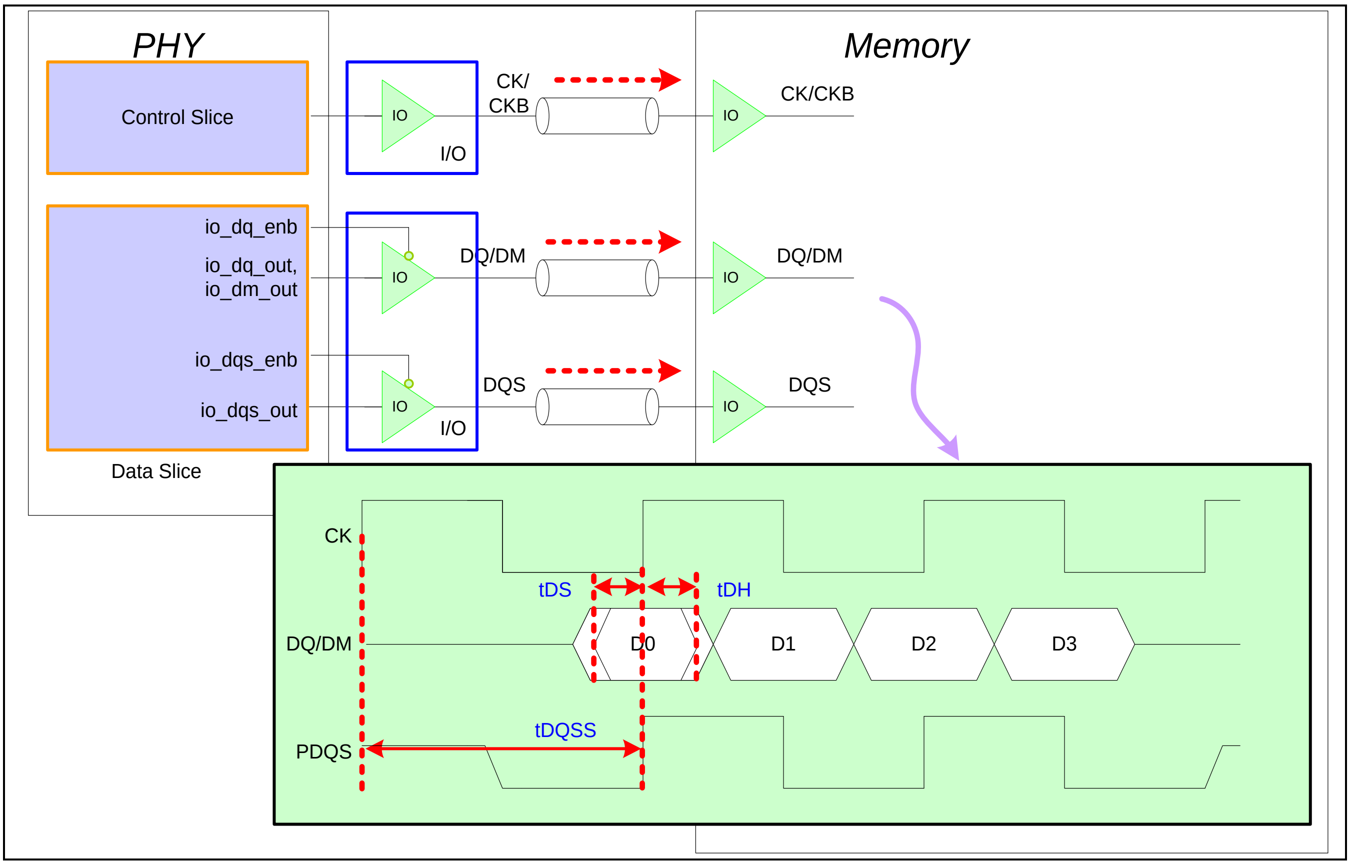

Figure 2-3. Block Diagram of the Write Path

2.2.2 Read Path

Read blocks capture valid data using strobe signal which come from memory for read transaction. DQs are 8-bit width data signals, DQS is strobe signal.

DQs and DQS from memory are edge aligned to CK edge. Therefore, to capture DQs using DQS, DQS should be delayed to make 90° phase shifted DQS, i.e. DQs should be center aligned to the edge of delayed DQS and captured DQs are stored in the FIFO.

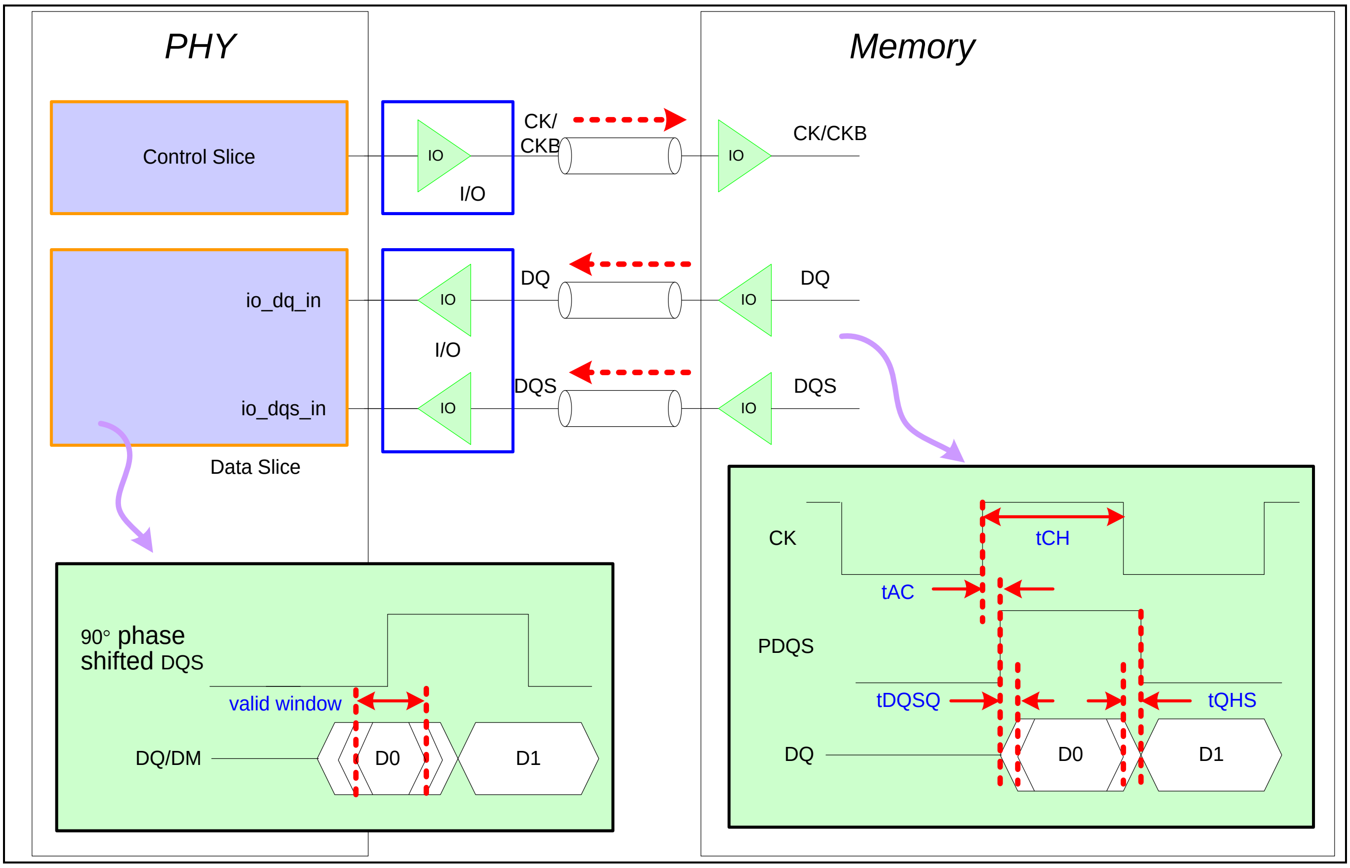

Figure 2-4. Block Diagram of the Read Path

2.3 ZQ CALIBRATION I/O

ZQ_IO calibrates the output and termination impedance and passes through impedance control signals to each I/O cells. (Refer to 8 Application Note). RZQ should be 240ohm.

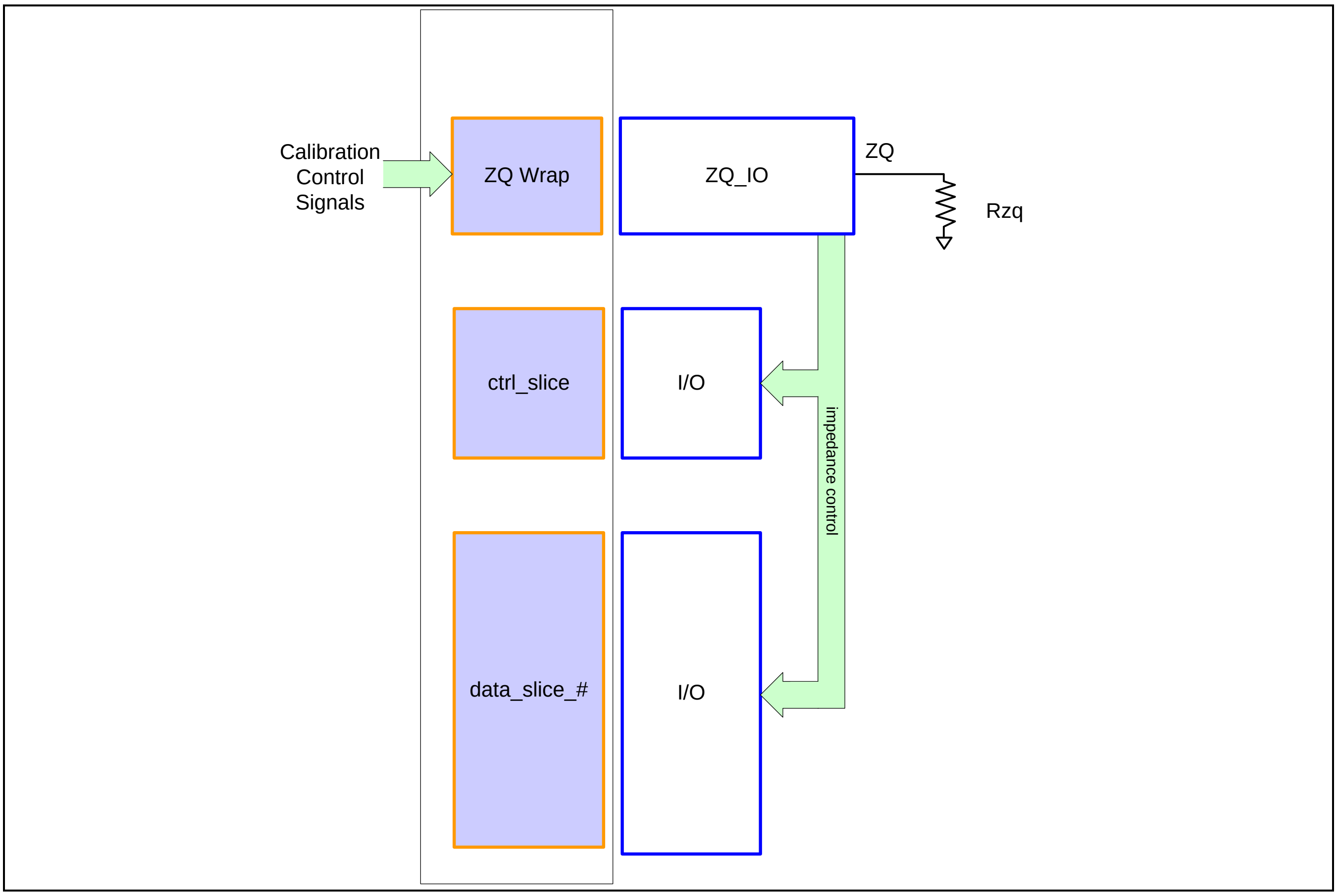

Figure 2-5. Connectivity Between ZQ I/O and the other I/Os

3 Signal Description

3.1 SYSTEM INTERFACE SIGNALS

Table 3-1. System Interface Signals

Name	Type	Description
clk2x	Input	0° phase system clock.
clk_en	Input	Clock Enable Signal to synchronize clk2x(PHY clock) with Controller Clock. Clock duty should be should be 49:51 ~ 51:49. ~~If the frequency ratio is 1:1, it should be always one.~~
clkm	Input	0° phase Master DLL clock (400~800MHz). This clock should be the same frequency clock with clk2x in normal mode and generated from the same PLL which clk2x is using. But Master DLL is not able to lock under 400MHz. if clk2x is under 400MHz, the double frequency of clk2x can be used for locking Master DLL for the low frequency operation. (p96) **NOTE:** All other input signals except for clkm should be generated in clk2x domain.
rst_n	Input	LOW active reset signal. In normal operation mode, reset is latched by clk and latched reset is used for rest of flipflops. This means that clock is required to release the reset of PHY. In scan mode, reset of flipflops are controlled by this pin directly.
mode_phy	Input	For normal operation, {mode_phy, mode_nand, mode_scan, mode_mux} should be 4'b0000. 0: Normal operation mode. 1: PHY test mode.
mode_nand	Input	For nand-tree test, {mode_phy, mode_nand, mode_scan, mode_mux} should be 4'b0100. 0: Normal operation mode. 1: Nand-Tree test mode.
mode_scan	Input	For scan test, {mode_phy, mode_nand, mode_scan, mode_mux} should be 4'b0010. 0: Normal operation mode. 1: Scan test mode.
mode_mux	Input	For full mux mode, mode_mux should be set. {mode_phy, mode_nand,mode_scan, mode_mux} should be 4'b0001. 1'b0: PHY is used. 1'b1: PHY is not used.
mode_highz	Input	If this bit is set, output is disabled.
mode_run[2:0]	Input	Valid only when mode_phy is set.

Preliminary product information describes products that are in development, for which full characterization data and associated errata are not yet available. Specifications and information herein are subject to change without notice.

		3'b000: I/O test mode. 3'b001: External PHY read feedback test mode. 3'b010: Internal PHY read feedback test mode. 3'b011: High-Z mode (All outputs are disabled). 3'b100: Internal PHY write feedback test mode.
ctrl_hcke	Input	This signal decides the reset value of CKE and some signals. If this bit is set, reset value of io_cke_out is 1. Otherwise, reset value of them is 0.
ctrl_pd[4:0]	Input	This field controls the input buffer of I/O. If this field is set, input buffer is turned-off for power down. This field should be 0 for normal operation.
ctrl_wake_up[4:0]	Input	This field controls toggling the input clock of 90 degree phase shift SDLL to save power. "ctrl_wakeup[4]" for ctrl_slice should be asserted 3 cycles prior to any command and then can be deasserted after the command is given to memory(The controller should issue NOP command during 3 cycles after ctrl_wake_up[4] is disabled). But if it doesn't use LPDDR2, "ctrl_wake_up[4]" can be always deasserted. "ctrl_wakeup[3:0]" for data_slice should be asserted 2 cycles prior to write command and then can be deasserted when ctrl_en falls. 5'b00000: All input clocks of Slave DLL isn't toggled. 5'b11111: All input clocks of Slave DLL is toggled.(Default) 5'b10000: The input clock of Slave DLL in ctrl_slice is toggled. But the input clock of SDLL in data_slice isn't toggled.

3.2 APB INTERFACE SIGNALS

Table 3-2. APB Interface Signals

Name	Type	Description
PCLK	Input	APB clock
PRESETn	Input	APB reset
PADDR[7:0]	Input	APB address
PSEL	Input	APB device select
PENABLE	Input	APB enable
PWRITE	Input	APB direction
PWDATA[31:0]	Input	APB write data
PRDATA[31:0]	Output	APB read data

3.3 ADDRESS & COMMAND INTERFACE SIGNALS

Table 3-3. Address & Command Interface Signals

Name	Type	Description
dfi_address_p0[19:0] dfi_address_p1[19:0]	Input	DFI address signals. These signals define the address information that is intended for the DRAM memory devices for all control commands. dfi_address_pN[15:0] for DDR2 and DDR3(dfi_address_pN[9:0] for LPDDR2) is used for address signals of rising edge of CK. dfi_address_pN[19:10] is used for address signals of falling edge of CK for LPDDR2.
dfi_reset_n_p0[1:0], dfi_reset_n_p1[1:0]	Input	DFI reset signal. Reset value for dfi_reset_n_pN[1:0] are 1. These signals are only required for DDR3 support. dfi_reset_n_p0/p1 is connected to io_reset_out.
dfi_cs_n_p0[1:0], dfi_cs_n_p1[1:0]	Input	DFI chip select bus.
dfi_bank_p0[2:0], dfi_bank_p1[2:0]	Input	DFI bank bus. These signals define the bank information.
dfi_ras_n_p0, dfi_ras_n_p1	Input	DFI row address strobe signal.
dfi_cas_n_p0, dfi_cas_n_p1	Input	DFI column address strobe signal.
dfi_we_n_p0, dfi_we_n_p1	Input	DFI write enable signal.
dfi_cke_p0[1:0], dfi_cke_p1[1:0]	Input	DFI clock enable signal.
dfi_odt_p0[1:0], dfi_odt_p1[1:0]	Input	DFI on-die termination control signal.
dfi_dram_clk_disable	Input	DRAM clock disable control. If this bit is HIGH, CK is set to LOW. Default should be high.

3.4 DATA INTERFACE SIGNALS

Table 3-4. Data Interface Signals

Name	Type	Description
dfi_wrdata_en_p0[1:0] dfi_wrdata_en_p1[1:0] (16-bit)/ dfi_wrdata_en_p0[3:0] dfi_wrdata_en_p1[3:0] (32-bit)	Input	Write data and data mask enable.
dfi_wrdata_p0[31:0], dfi_wrdata_p1[31:0](16-bit)/ dfi_wrdata_p0[63:0], dfi_wrdata_p1[63:0](32-bit)	Input	Write data bus.
dfi_wrdata_mask_p0[3:0] dfi_wrdata_mask_p0[3:0] (16-bit)/ dfi_wrdata_mask_p0[7:0] dfi_wrdata_mask_p0[7:0] (32-bit)	Input	Write data byte mask.
dfi_rddata_en_p0[1:0], dfi_rddata_en_p1[1:0] (16-bit)/ dfi_rddata_en_p0[3:0] dfi_rddata_en_p1[3:0] (32-bit)	Input	Read data enable. Active HIGH signal to enable FIFO read. After PHY detects HIGH dfi_rddata_en_p* at rising edge system clock, data in FIFO appear on dfi_rdata_w* port after 2 cycles.
dfi_rddata_valid_w0[1:0], dfi_rddata_valid_w1[1:0] (16-bit)/ dfi_rddata_valid_w0[3:0], dfi_rddata_valid_w1[3:0] (32-bit)	input	Read data valid indicator. The dfi_rddata_valid_w* signal will be asserted with the read data for the number of cycles that data is being sent. The timing is the same as for the dfi_rddata_w* bus.
dfi_rdata_w0[31:0], dfi_rdata_w1[31:0] (16-bit)/ dfi_rdata_w0[63:0], dfi_rdata_w1[63:0] (32-bit)/	Output	Read data bus. Read data is expected to be received at the Controller within Tphy_rdlat cycles after the dfi_rddata_en* signal is asserted.
ctrl_read_p0[1:0] ctrl_read_p1[1:0] (16-bit)/ ctrl_read_p0[3:0] ctrl_read_p1[3:0] (32-bit)	Input	Active HIGH signal to turn on termination for memory read. If this field is HIGH, termination resistors of the I/Os are turned-on. This field should be carefully controlled to reduce the power consumption of I/Os for DDR2/DDR3. (refer to 8.6.2) It's highly recommended for each bit to be controlled by the different corresponding filpflop. DDR, LPDDR, LPDDR2 : 1'b0 DDR2, DDR3 : always set LOW except for data read.
ctrl_gate_p0[3:0], ctrl_gate_p1[3:0]	Input	DQS clean signal.

3.5 UPDATE, STATUS AND TRAINING INTERFACE SIGNALS

Table 3-5. Update, Status and Training Interface Signals

Name	Type	Description
dfi_ctrlupd_req	Input	MC-initiated update request.
dfi_ctrlupd_ack	Output	MC-initiated update acknowledge.
dfi_dram_clk_disable	Input	DRAM clock disable. When active, this indicates to the PHY that the clocks to the DRAM devices must be disabled such that the clock signals hold a constant value. When the dfi_dram_clk_disable signal is inactive, the DRAMs should be clocked normally.
dfi_freq_ratio	Input	DFI frequency ratio indicator. This field should be tied by "2'b01". ~~`b00 = 1:1 MC:PHY frequency ratio (matched frequency)~~ `b01 = 1:2 MC:PHY frequency ratio
dfi_init_start	Input	During initialization, it is an indication to the PHY that all configurations for PHY have been defined. During normal operation, PHY doesn't support Frequency Change Protocol by asserting "dfi_init_complete". Please refer to 8.2for Frequency Change.
dfi_init_complete	Output	PHY initialization complete.
~~dfi_rdlvl_mode~~	~~Output~~	~~2'b10 : PHY Evaluation mode.~~
dfi_rdlvl_en	Input	DFI Read Leveling Enable signal. During dfi_rdlvl_en=1, DLL Force mode should be used. ctrl_force should be equal to "lock_value[9:2]" before ctrl_dll_on=0.
dfi_rdlvl_gate_en	Input	DFI Gate leveling Enable signal. The dfi_init_complete signal indicates that the PHY is able to respond to any proper stimulus on the DFI. "dfi_rdlvl_en" should be also enabled for gate leveling. And gate leveling for LPDDR2 and LPDDR3 is not guaranteed due to too much variation of tDQSCK.
dfi_rdlvl_resp	Output	Read leveling response. The response indicates that the PHY has completed data eye training or gate training and centered the DQS relative to the data or placed the gate within the DQS preamble.

Preliminary product information describes products that are in development, for which full characterization data and associated errata are not yet available. Specifications and information herein are subject to change without notice.

3.7 SSTL I/O INTERFACE SIGNALS

Table 3-6. SSTL I/O Interface Signals

Name	Type	Description
io_adct_out[15:0]	Output	Address. These I/Os are used for ADDR
io_adct_en[15:0]	Output	Address I/O output enable
io_bank_out[2:0]	Output	Bank Address
io_bank_en[2:0]	Output	Bank Address I/O output enable
io_reset_out	Output	RESET for DDR3 Memory
io_reset_en	Output	RESET I/O output enable
io_odt_out[1:0]	Output	ODT
io_odt_en[1:0]	Output	ODT I/O output enable
io_cs_out[1:0]	Output	CS
io_cs_en[1:0]	Output	CS I/O output enable
io_ras_out	Output	RAS, row address selection.
io_ras_en	Output	RAS I/O output enable
io_cas_out	Output	CAS, column address selection.
io_cas_en	Output	CAS I/O output enable
io_wen_out	Output	WE, write enable.
io_wen_en	Output	WE I/O output enable
io_cke_out[1:0]	Output	CKE, clock enable.
io_cke_en[1:0]	Output	CKE I/O output enable
io_ck_out[1:0]	Output	CK/CKB differential clock
io_ck_en[1:0]	Output	CK I/O output enable
io_ckb_en[1:0]	Output	CKB I/O output enable
io_adct_pd[15:0]	Output	Address I/O Receiver Power Down Enable.
io_bank_pd[2:0]	Output	Bank Address I/O Receiver Power Down Enable.
io_reset_pd	Output	RESET I/O Receiver Power Down Enable.
io_odt_pd[1:0]	Output	ODT I/O Receiver Power Down Enable.
io_cs_pd[1:0]	Output	CS I/O Receiver Power Down Enable.
io_ras_pd	Output	RAS I/O Receiver Power Down Enable.
io_cas_pd	Output	CAS I/O Receiver Power Down Enable.
io_wen_pd	Output	WE I/O Receiver Power Down Enable.
io_ck_pd	Output	CK I/O Receiver Power Down Enable.
io_cke_pd[1:0]	Output	CKE I/O Receiver Power Down Enable.
io_adct_in[15:0]	Input	Address Input from I/O.
io_bank_in[2:0]	Input	Bank Address Input from I/O.
io_reset_in	Input	RESET Input from I/O.
io_odt_in[1:0]	Input	ODT Input from I/O.

Name	Type	Description
io_cs_in[1:0]	Input	CS Input from I/O.
io_ras_in	Input	RAS Input from I/O.
io_cas_in	Input	CAS Input from I/O.
io_wen_in	Input	WE Input from I/O.
io_ck_in	Input	CK Input from I/O.
io_cke_in[1:0]	Input	CKE Input from I/O.
io_dq_in[15:0] (16-bit)/ io_dq_in[31:0] (32-bit)	Input	DQ, Input Data.
io_dq_out[15:0] (16-bit)/ io_dq_out[31:0] (32-bit)	Output	DQ, Output Data.
io_dq_en[15:0] (16-bit)/ io_dq_en[31:0] (32-bit)	Output	DQ I/O output enable
io_dq_pdn[31:0]	Output	DQ I/O Pull-down enable
io_dq_pup[31:0]	Output	DQ I/O Pull-up enable
io_dq_read[31:0]	Output	DQ I/O Drive tri-state enable
io_dq_cmosrcv[31:0]	Output	DQ I/O CMOS/Differential receiver selection
io_dq_pd[31:0]	Output	Power down pin for DQ I/O receiver
io_dm_out[1:0] (16-bit)/ io_dm_out[3:0] (32-bit)	Output	DM, Data mask.
io_dm_en[3:0]	Output	DM I/O output enable
io_dm_in[3:0]	Output	DM I/O Input Data
io_dm_pd[3:0]	Output	Power down for DM I/O receiver
io_pdqs_out[1:0] (16-bit)/ io_pdqs_out[3:0] (32-bit)	Output	Positive DQS.
io_pdqs_en[1:0] (16-bit)/ io_pdqs_en[3:0] (32-bit)	Output	Positive DQS output enable
io_pdqs_pdn[3:0]	Output	Positive DQS Pull-down enable
io_pdqs_pup[3:0]	Output	Positive DQS Pull-up disable
io_pdqs_read[3:0]	Output	Positive DQS Drive tri-state enable
io_ndqs_en[1:0] (16-bit)/ io_ndqs_en[3:0] (32-bit)	Output	Negative DQS output enable.
Io_ndqs_pdn[3:0]	Output	Negative DQS Pull-down enable
io_ndqs_pup[3:0]	Output	Negative DQS Pull-up disable
io_ndqs_read[3:0]	Output	Negative DQS Drive tri-state enable
io_adct_dds[47:0]	Output	Address Driver Strength Control
io_bank_dds[8:0]	Output	Bank Address Driver Strength Control
io_reset_dds[2:0]	Output	RESET Driver Strength Control
io_odt_dds[5:0]	Output	ODT Driver Strength Control
io_cs_dds[5:0]	Output	CS Driver Strength Control

Name	Type	Description
io_ras_dds[2:0]	Output	RAS Driver Strength Control
io_cas_dds[2:0]	Output	CAS Driver Strength Control
io_wen_dds[2:0]	Output	WE Driver Strength Control
io_ck_dds[2:0]	Output	CK Driver Strength Control
io_cke_dds[5:0]	Output	CKE Driver Strength Control
io_dq_dds[15:0] (16-bit)/ io_dq_dds[95:0] (32-bit)	Output	DQ, Driver Strength Control
io_dm_dds[11:0]	Output	DM Driver Strength Control
io_dqs_dds[11:0]	Output	DQS Driver Strength Control
io_zq_clk	Output	Clock for impedance calibration block
io_zq_reset	Output	Reset signal for initialization (Low → High) Low : Reset mode High : Active mode
io_zq_ack	Input	Calibration finish indication signal(High: calibration is finished)
io_zq_err	Input	Calibration fail indication signal(High: calibration failed)
io_zq_pmon[2:0]	Input	Calibrated pull up control bits sent to core, accessed on completion of calibration cycle
io_zq_nmon[2:0]	Input	Calibrated pull down control bits sent to core, accessed on completion of calibration cycle
io_zq_req_force	Output	Handshake signal to request force calibration from the external code
io_zq_req_long	Output	Handshake signal to request long calibration
io_zq_req_short	Output	Handshake signal to request short calibration
io_zq_force_impp[2:0]	Output	Initial code for pull up
io_zq_force_impn[2:0]	Output	Initial code for pull down
io_zq_mode_rgddr3	Output	GDDR3 mode enable signal(High: GDDR3 mode)
io_zq_mode_noterm	Output	Control pin to enable ODT(Low: ODT enabled)
io_zq_mode_term[2:0]	Output	On-die termination select, receive mode
io_zq_mode_dds[2:0]	Output	Control pins to change driver's strength. It is connected to ZQ I/O.

3.8 DIRECT ACCESS INTERFACE SIGNALS

Table 3-7. Direct Access Interface Signals

Name	Type	Description
ctrl_io_adct[15:0]	Input	When it's mux mode, ctrl_io_adct[15:0] controls io_adct_out.[15:0].
ctrl_io_bank[2:0]	Input	When it's mux mode, ctrl_io_bank[2:0] controls io_bank_out.[2:0].
ctrl_io_reset	Input	When it's mux mode, ctrl_io_reset controls io_reset_out.
ctrl_io_odt[1:0]	Input	When it's mux mode, ctrl_io_odt[1:0] controls io_odt_out.[1:0].
ctrl_io_cs[1:0]	Input	When it's mux mode, ctrl_io_cs[1:0] controls io_cs_out.[1:0].
ctrl_io_ras	Input	When it's mux mode. This controls io_ras_out. If not used, tie to zero.
ctrl_ io_cas	Input	When it's mux mode. This controls io_cas_out. If not used, tie to zero.
ctrl_ io_wen	Input	When it's mux mode. This controls io_wen_out. If not used, tie to zero.
ctrl_ io_cke[1:0]	Input	When it's mux mode. This controls io_cke_out. If not used, tie to zero.
ctrl_io_ck	Input	When it's mux mode. This controls io_ck_out. If not used, tie to zero.
ctrl_io_data_en[31:0]	Input	When it's bypass/mux mode, io_dq_en[31:0] is controlled by this field. If bypass mode is not used, tie to 0.
ctrl_io_dm_en[3:0]	Input	When it's bypass/mux mode, io_dm_en[3:0] is controlled by this field. If bypass mode is not used, tie to 0.
ctrl_io_dqs_en[3:0]	Input	When it's bypass/mux mode, io_pdqs_en[3:0] is controlled by this field. If bypass mode is not used, tie to 0.
ctrl_io_wdata[15:0](16-bit) ctrl_io_wdata[31:0](32-bit)	Input	When it's bypass/mux mode, io_dq_out is controlled by this field. If bypass mode is not used, tie to 0.
ctrl_io_rdata[15:0](16-bit) ctrl_io_rdata[31:0](32-bit)	Output	When it's bypass/mux mode, ctrl_io_rdata is controlled by io_dq_in. If it's not bypass/mux mode, this field is not valid.
ctrl_io_wdqs[1:0](16-bit) ctrl_io_wdqs[3:0](32-bit)	Input	When it's bypass/mux mode, io_dqs_out is controlled by this field. If bypass mode is not used, tie to 0.
ctrl_io_rdqs[1:0](16-bit) ctrl_io_rdqs[3:0](32-bit)	Output	When it's bypass/mux mode, ctrl_io_rdqs is controlled by io_dqs_in. If it's not bypass/mux mode, this field is not valid.
ctrl_io_rndqs[1:0](16-bit) ctrl_io_rndqs[3:0](32-bit)	Output	When it's bypass/mux mode, ctrl_io_rndqs is controlled by io_ndqs_in. If it's not bypass/mux mode, this field is not valid.
ctrl_io_dm[1:0](16-bit) ctrl_io_dm[3:0](32-bit)	Input	When it's bypass/mux mode, io_dm_out is controlled by this field. If bypass mode is not used, tie to 0.
ctrl_io_rdm[1:0](16-bit) ctrl_io_rdm[3:0](32-bit)	Input	When it's bypass/mux mode, ctrl_io_rdm is controlled by io_dm_in. If bypass mode is not used, tie to 0.
ctrl_io_adct_in[15:0]	Output	When it's mux mode, ctrl_io_adct[15:0] is controlled by io_adct_in.[15:0].
ctrl_io_bank_in[2:0]	Output	When it's mux mode, ctrl_io_bank[2:0] is controlled by io_bank_in[2:0].
ctrl_io_reset_in	Output	When it's mux mode, ctrl_io_reset is controlled by io_reset_in.
ctrl_io_odt_in[1:0]	Output	When it's mux mode, ctrl_io_odt[1:0] is controlled by io_odt_in[1:0].
ctrl_io_cs_in[1:0]	Output	When it's mux mode, ctrl_io_cs[1:0] is controlled by io_cs_in[1:0].

Name	Type	Description
ctrl_io_ras_in	Output	When it's mux mode. This is controlled by io_ras_in. If not used, tie to zero.
ctrl_ io_cas_in	Output	When it's mux mode. This is controlled by io_cas_in. If not used, tie to zero.
ctrl_ io_wen_in	Output	When it's mux mode. This is controlled by io_wen_in. If not used, tie to zero.
ctrl_ io_cke_in[1:0]	Output	When it's mux mode. This is controlled by io_cke_in. If not used, tie to zero.
ctrl_io_ck_in	Output	When it's mux mode. This is controlled by io_ck_in. If not used, tie to zero.
ctrl_io_adct_en[15:0]	Input	When it's mux mode, ctrl_io_adct_en[15:0] controls io_adct_en[15:0].
ctrl_io_bank_en[2:0]	Input	When it's mux mode, ctrl_io_bank_en[2:0] controls io_bank_en[2:0].
ctrl_io_reset_en	Input	When it's mux mode, ctrl_io_reset_en controls io_reset_en.
ctrl_io_odt_en[1:0]	Input	When it's mux mode, ctrl_io_odt_en[1:0] controls io_odt_en[1:0].
ctrl_io_cs_en[1:0]	Input	When it's mux mode, ctrl_io_cs_en[1:0] controls io_cs_en[1:0].
ctrl_io_ras_en	Input	When it's mux mode. This controls io_ras_en. If not used, tie to zero.
ctrl_ io_cas_en	Input	When it's mux mode. This controls io_cas_en. If not used, tie to zero.
ctrl_ io_wen_en	Input	When it's mux mode. This controls io_wen_en. If not used, tie to zero.
ctrl_ io_cke_en[1:0]	Input	When it's mux mode. This controls io_cke_en. If not used, tie to zero.
ctrl_io_ck_en	Input	When it's mux mode. This controls io_ck_en. If not used, tie to zero.

NOTE: MUX mode can be used for JTAG. If not using JTAG(=MUX mode), All inputs of "ctrl_io_*" should be tied to zero and All output of "ctrl_io_*" should be floating.

3.9 FEEDBACK TEST INTERFACE SIGNALS

Table 3-8. Feedback Test Interface Signals

Name	Type	Description
test_ext_dfdqs	Input	1'b0 : single-ended DQS 1'b1 : differential DQS (DDR2 and LPDDR2)
test_ext_cmosrcv	Input	This field controls the input mode of I/O 1'b0 : Differential receiver mode for high speed operation 1'b1 : CMOS receiver mode for low speed operation (< 200MHz)
test_ext_lpddr2	Input	1'b0 : DDR2 and DDR3 1'b1 : LPDDR2
test_ext_locked	Output	DLL stable lock information. This field is set after test_ext_flock is set. This field is required for stable lock status check.
test_ext_ref[3:0]	Input	test_ext_ref[0] controls "byte_level_en"(PHY_CON[13]) during feedback test.
test_ext_dll_on	Input	External DLL on signal to turn on the DLL for test mode. This signal should be kept HIGH for test mode. If this signal becomes LOW, DLL is turned off and the lock value will be the maximum for supporting low frequency at External Feedback Test. This bit should be kept set before test_ext_start is set to turn on the DLL.
test_ext_start_point[6:0]	Input	External DLL control signal for test mode. Initial DLL lock start point. This is the number of delay cells and is the start point where "DLL" start tracing to be locked.
test_ext_inc[6:0]	Input	External DLL control signal for test mode. Increase amount of start point.
test_ext_clock	Output	External status signal for DLL for test mode. Coarse lock information.
test_ext_flock	Output	External status signal for DLL for test mode. Fine lock information.
test_ext_lock_value[8:0]	Output	External status signal for DLL for test mode. Locked delay line encoding value. Test_ext_lock_value[8:2] : number of delay cells for coarse lock. Test_ext_lock_value[1:0] : control value for fine lock.
test_ext_offsetd[7:0]	Input	External control signal for test. Offset amount for 270° clock generation for control path. If this field is fixed, this should not be changed during operation. This value is valid only after an update request. Test_ext_offsetd[7] = 1 : (tFS : fine step delay) 270° delay amount – test_ext_offsetd[6:0] x tFS test_ext_offsetd[7] = 0 : 270° delay amount + test_ext_offsetd[6:0] x tFS
test_ext_shiftc[4:0]	Input	External control signal for test. GATEin signal delay amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid only after an update request. This value is limited by the half of the maximum delay in Master Delay Line. GATEin signal will be delayed by T/16 whenever it increases one. 00000: 0(0° shift), 00001: T/16(22.5° shift), 00010: T/8(45° shift) , 00011: 3*T/16(67.5° shift)

Name	Type	Description
		00100: T/4(90° shift) , 00101: 5*T/16(112.5° shift) … (Increased by T/16) 11110: T+15*T/16(697.5° shift) , 10111: 2T(720° shift)
test_ext_offsetc[7:0]	Input	External control signal for test. If this field is fixed, this should not be changed during operation. This value is valid only after an update request. gate signal offset amount : test_ext_offsetc[7] = 1 : (tFS : fine step delay) 90° delay amount – ctrl_offsetc[6:0] x tFS test_ext_offsetc[7] = 0 : 90° delay amount + ctrl_offsetc[6:0] x tFS
test_ext_offsetr[7:0]	Input	External control signal for test. If this field is fixed, this should not be changed during operation. This value is valid only after an update request. Read path(0/1/2/3) offset amount : test_ext_offsetr[7] = 1 : (tFS : fine step delay) 90° delay amount – ctrl_offsetr[6:0] x tFS test_ext_offsetr[7] = 0 : 90° delay amount + ctrl_offsetr[6:0] x tFS
test_ext_offsetw[7:0]	Input	External control signal for test. If this field is fixed, this should not be changed during operation. This value is valid only after an update request. Write path(0/1/2/3) offset amount : test_ext_offset[7] = 1 : (tFS : fine step delay) DQ 270° delay amount – ctrl_offset[6:0] x tFS test_ext_offset[7] = 0 : DQ 270° delay amount + ctrl_offset[6:0] x tFS
test_start[2:0] (16-bit)/ test_start[4:0] (32-bit)	Input	Feedback test start signal for each slice.
test_resync	Input	External control active HIGH signal for test mode. This signal should become LOW after set HIGH for normal operation. Before set and clear this signal, test_ext_locked should be checked.
test_err[2:0] (16-bit)/ test_err[4:0] (32-bit)	Output	Feedback test stop with error for each slice.
test_oky[2:0] (16-bit)/ test_oky[4:0] (32-bit)	Output	Feedback test completion without error for each slice.
test_ext_mode[3:0]	Input	External control signal for test.(default:4'b0000) test_ext_mode[1:0] = 2'b00 : The value of rdlvl_offsetr* will be on "test_ext_rdlvl_vwml" test_ext_mode[1:0] = 2'b01 : The value of rdlvl_offsetw* will be on "test_ext_rdlvl_vwml" test_ext_mode[1:0] = 2'b10 : The complete signal of Read Levleing will be on on "test_ext_rdlvl_vwml[0],[8],[16],[24] test_ext_mode[2] : 1'b0(Read Feedback), 1'b1(Write Feedback) test_ext_mode[3] : 1'b0(DLL Enable), 1'b1(DLL Disable),

Name	Type	Description
test_ext_init_complete	Output	dfi_init_complete signal output during feedback test.
test_ext_rdlvl_en	Input	External control signal for test.(default: 1'b0)
test_ext_rdlvl_wr_en	Input	External control signal for test.(default: 1'b0)
test_ext_gatelvl_en	Input	External control signal for test.(default: 1'b0)
test_ext_rdlvl_incr_adj[3:0]	Input	It decides how many delay steps will be used during training(default: 4'h4)
test_ext_rdlvl_vwmc	Output	"Valid Window Margin"(VWM) after Read Leveling will be defined. It will be the center position in VWM.
test_ext_rdlvl_vwml	Output	"Valid Window Margin"(VWM) after Read Leveling will be defined. It will be the left position in VWM.
test_ext_rdlvl_vwmr	Output	"Valid Window Margin"(VWM) after Read Leveling will be defined. It will be the right position in VWM.

These test mode signals are effective only when PHY is in the feedback test mode. When mode_phy is set and mode_run[2:0] is 3'b001, 3'b010 or 3’b100, DLL and Feedback test path can be controlled by these external signals.

Refer to section 7 for the information to assign test signals to external pins.

3.10 I/O TEST INTERFACE SIGNALS

Table 3-9. I/O Test Interface Signals

Name	Type	Description
test_ext_en	Input	Direct I/O enable control signal. If this field is high, I/O output is enabled.
test_ext_out	Input	Direct I/O output control signal. This field controls I/O output value.
test_ext_read	Input	Direct I/O termination control signal. This field controls I/O termination resistor.

Caution: Before testing VOL/VOH and termination resistor of I/O, ZQ calibration should be done.

These test mode signals are effective only when run_mode is external I/O test mode. When mode_phy is set and mode_run is 3'b000, all output signals to I/Os can be controlled by these external signals.

- When I/O test mode :
  - {test_ext_en, test_ext_read, test_ext_out} = 3'b00X : Output is high-Z.(Except for PDQS,NDQS,DQ)
  - {test_ext_en, test_ext_read, test_ext_out} = 3'b100 : Output is low.
  - {test_ext_en, test_ext_read, test_ext_out} = 3'b101 : Output is high.
  - {test_ext_en, test_ext_read, test_ext_out} = 3'bX1X : Termination resistor is on.

- When high-z mode:
  - {test_ext_en, test_ext_out} = 2'b0X : outputs for DQ and PDQS/NDQS are disabled

3.11 SCAN TEST INTERFACE SIGNALS

Table 3-10. SCAN Test Interface Signals

Name	Type	Description
test_si[9:0] (16-bit)/ test_si[19:0] (32-bit)/	Input	SCAN input signals for each scan-chain.(32-bit PHY) This field is 5-bit for 16-bit PHY.
test_so[9:0] (16-bit)/ test_so[19:0] (32-bit)/	Output	SCAN output signals for each scan-chain.(32-bit PHY) This field is 5-bit for 16-bit PHY.
test_se	Input	SCAN enable signal.

4 Register Description

4.1 Register Overview

Table 4-1. Register Overview

Register	Offset	R/W	Description	Initial Value
PHY_CON0	0x00	R/W	PHY Control Register 0	0x1702_0A40
PHY_CON1	0x04	R/W	PHY Control Register 1	0x0921_0100
PHY_CON2	0x08	R/W	PHY Control Register 2	0x0001_0004
PHY_CON3	0x0C	R/W	GATE SDLL Code Control Register 0	0x0021_0842
PHY_CON4	0x10	R/W	READ SDLL Code Control Register 0	0x0808_0808
PHY_CON5	0x14	R/W	READ Mode Control Register	--
PHY_CON6	0x18	R/W	WRITE SDLL Code Control Register 0	0x0808_0808
~~PHY_CON7~~	~~0x1C~~	~~R/W~~	~~WRITE SDLL Code Control Register 1~~	~~0x0000_0008~~
PHY_CON8	0x20	R/W	GATE SDLL Code Control Register 1	0x0000_0000
~~PHY_CON9~~	~~0x24~~	~~R/W~~	~~GATE SDLL Code Control Register 2~~	~~0x0000_0000~~
PHY_CON10	0x28	R/W	CMD SDLL Code Control Register	0x0000_0008
PHY_CON11	0x2C	R/W	--	0x0000_0000
PHY_CON12	0x30	R/W	MDLL Control Register 0	0x1010_0070
PHY_CON13	0x34	R	MDLL Control Register 1	---
PHY_CON14	0x38	R/W	Low Power Control Register	0x001F_0000
PHY_CON15	0x3C	R/W	Feedback Test Control Register	0x0000_0000
PHY_CON16	0x40	R/W	ZQ Control Register	0x0800_0304
PHY_CON17	0x48	R	ZQ Status Register	---
PHY_CON18	0x4C	R	Read Leveling Status Register 0	---
PHY_CON19	0x50	R	Read Leveling Status Register 1	---
PHY_CON20	0x54	R	Read Leveling Status Register 2	---
PHY_CON21	0x58	R	Read Leveling Status Register 3	---
PHY_CON22	0x5C	R/W	Read Leveling Control Register 0	0x0000_0208
PHY_CON23	0x60	R/W	Read Leveling Control Register 1	0x0000_03FF
PHY_CON24	0x64	R/W	Read Leveling Control Register 2	0x0000_0000
PHY_CON25	0x68	R/W	Read Leveling Control Register 3	0x105E_107E
PHY_CON26	0x6C	R/W	Read Leveling Control Register 4	0x0008_107F
PHY_CON27	0x70	R	PHY Control Register 27	---
PHY_CON28	0x74	R	Read Leveling Status Register 4	---

Register	Offset	R/W	Description	Initial Value
PHY_CON29	0x78	R	Version Information Register	0x0500_0390
PHY_CON30	0x7C	R/W	Write Leveling Control Register	0x0000_0000
PHY_CON31	0x80	R/W	CA Deskew Control Register 0	0x0000_0000
PHY_CON32	0x84	R/W	CA Deskew Control Register 1	0x0000_0000
PHY_CON33	0x88	R/W	CA Deskew Control Register 2	0x0000_0000
PHY_CON34	0x8C	R/W	CA Deskew Control Register 3	0x0000_0000
PHY_CON37	0x98	R/W	CMD Deskew Control Register 1	0x0000_0000
PHY_CON39	0xA0	R/W	Driver Strength Control Register	0x0000_0000
PHY_CON40	0xA4	R/W	ZQ Divider Control Register	0x0000_0007
PHY_CON41	0xA8	R/W	ZQ Timer Control Register	0x0000_00F0
PHY_CON42	0xAC	R/W	PHY Control Register 3	0x0000_0000

4.2 PHY Control Register

Table 4-2. PHY_CON0(Address Offset=0x00)

Field	Bit	R/W	Description	Initial Value
Reserved	[31:29]		Should be zero	0x0
T_WrWrCmd	[28:24]	R/W	It controls the interval between Write and Write during DQ Calibration. This value should be always kept by 5'h17. It will be used for debug purpose.	5'h17
ctrl_upd_range	[21:20]	R/W	It decides how many differences between the new lock value and the current lock value which is used in Slave-DLL is needed for updating lock value. 2'b00 : To update always 2'b01 : To ignore the lower 1 bit in ctrl_lock_value 2'b10 : To ignore the lower 2 bits in ctrl_lock_value 2'b11 : To ignore the lower 3 bits in ctrl_lock_value	2'b00
T_WrRdCmd	[19:17]	R/W	It controls the interval between Write and Read by cycle unit during Write Calibration. It will be used for debug purpose. 3'b111 : tWTR = 6 cycles 3'b110 : tWTR = 4 cycles	3'b001
ctrl_wrlvl_en (=wrlvl_mode)	[16]	R/W	Write Leveling Enable.	1'b0
p0_cmd_en	[14]	R/W	1'b0 : Issue Phase1 Read Command during read leveling 1'b1 : Issue Phase0 Read Command during read leveling	1'b0
byte_rdlvl_en	[13]	R/W	Byte Read Leveling enable. It should be set if memory supports toggling only 1 DQ bit except for other 7 bits during read leveling.	1'b0
ctrl_ddr_mode	[12:11]	R/W	2’b00: DDR2 and LPDDR1 2’b01: DDR3 2'b10: LPDDR2 2'b11: LPDDR3	2'b11

Field	Bit	R/W	Description	Initial Value
ctrl_dfdqs	[9]	R/W	1'b0: single-ended DQS 1'b1: differential DQS	1'b1
ctrl_shgate	[8]	R/W	This field controls the gate control signal 1'b0: gate signal length = "burst length / 2" + N (DQS Pull-Down mode, ctrl_pulld_dqs[3:0] == 4'b1111, N = 0,1,2…) 1'b1: gate signal length = "burst length / 2" – 1	1'b0
ctrl_ckdis	[7]	R/W	This field controls the CK/CKB 1"b0: Clock output is enabled 1"b1: Clock output is disabled	1'b0
ctrl_atgate	[6]	R/W	If ctrl_atgate=0, Controller should generate ctrl_gate_p*, ctrl_read_p*. If ctrl_atgate=1, PHY will generate ctrl_gate_p*, ctrl_read_p*, but it has some constraints. This setting can be supported only over RL=4, BL and RL should be properly set to operate with ctrl_atgate=1.	1'b1
ctrl_read_disable	[5]	R/W	Read ODT(On-Die-Termination) Disable Signal. This signal should be one in LPDDR2 or LPDDR3 mode. 1'b1 : drive ctrl_read_p* to 0.(If using LPDDR2 or LPDDR3, ctrl_read_p* will be always 0 by enabling this field.) 1'b0 : drive ctrl_read_p* normally.	1'b0
ctrl_cmosrcv	[4]	R/W	This field controls the input mode of I/O 1'b0: Differential receiver mode for high speed operation 1'b1: CMOS receiver mode for low speed operation (< 200MHz)	1'b0
ctrl_read_width	[3]	R/W	The default is 1'b0. We strongly recommend that it should have one more idle cycle between read and write because the variation of data arrival time during read can be greater than 1 cycle. If it is 1'b1, the package and board should be carefully optimized with a short length and it should be used at the low frequency. Please refer to Figure 8-6. 1'b0: Termination on period is (BL/2+1.5) cycle (Default) 1'b1: Termination on period is (B/2+1) cycle(Not recommended)	1'b0
ctrl_fnc_fb	[2:0]	R/W	Valid only when {mode_phy, mode_nand, mode_scan, mode_mux} is 4'b00000. For normal operation 3'b000: Normal operation mode. For ATE test purpose 3'b010: External FNC read feedback test mode. 3'b011: Internal FNC read feedback test mode. For Board test purpose 3'b100: External PHY read feedback test mode. When memory is not attached on the board 3'b101: Internal PHY read feedback test mode. mode_highz should be set. 3'b110: Internal PHY write feedback test mode. mode_highz should be set. For Power Down	3'b000

Field	Bit	R/W	Description	Initial Value
			3'b111: Power Down Mode for SSTL I/O	

Table 4-3. PHY_CON1(Address Offset=0x04)

Field	Bit	R/W	Description	Initial Value
ctrl_gateadj	[31:28]	R/W	It adjusts the enable time of "ctrl_gate" on a clock cycle base. MSB(bit3) controls direction: 1'b1(subtract delay), 1'b0(add delay), Bit[2:0] set delay value	4'h0
ctrl_readadj	[27:24]	R/W	It adjusts the enable time of "ctrl_read" on a clock cycle base. MSB(bit3) controls direction: 1'b1(subtract delay), 1'b0(add delay), Bit[2:0] set delay value	4'h9
ctrl_gateduradj	[23:20]	R/W	It adjusts the duration cycle of "ctrl_gate" on a clock cycle base. MSB(bit3) controls direction: 1'b1(subtract delay), 1'b0(add delay), Bit[2:0] set delay value	4'h2
rdlvl_pass_adj	[19:16]	R/W	This field controls how many times "Read" should be operated well to determine if it goes into VWP(Valid Window Period) or not. (default: 4'h1)	4'h1
rdlvl_rddata_adj	[15:0]	R/W	It decides the pattern to be read during read or write calibration. (default: 16'h0100) 16'h0100 : DDR3 ("byte_rdlvl_en=1") 16'h0001 : LPDDR3 ("byte_rdlvl_en=1")	16'h0100

Table 4-4. PHY_CON2(Address Offset=0x08)

Field	Bit	R/W	Description	Initial Value
ctrl_readduradj	[31:28]	R/W	It adjusts the duration cycle of "ctrl_read" on a clock cycle base. MSB(bit3) controls direction: 1'b1(subtract delay), 1'b0(add delay), Bit[2:0] set delay value	0x0
wr_deskew_en (=wr_cal_start)	[27]	R/W	DQ Calibration Start Signal to align DQ, DM during write.	1'b0
wr_deskew_con (=wr_cal_mode)	[26]	R/W	If it is enabled, PHY will use "Write Slave DLL Code" which has got during Read Leveling.	1'b0
rdlvl_en (=rd_cal_mode)	[25]	R/W	When rd_cal_mode=1, Read leveling offset values will be used instead of ctrl_offsetr*. If read leveling is used, this value should be high during operation.	1'b0
rdlvl_gate_en (=gate_cal_mode)	[24]	R/W	When gate_cal_mode=1, Gate leveling offset value will be used instead of ctrl_shiftc*. If gate leveling is used, this value should be high during operation.	1'b0
rdlvl_ca_en (=ca_cal_mode)	[23]	R/W	When ca_cal_mode=1, CA Calibration offset value will be used and updated.	1'b0
rdlvl_incr_adj	[22:16]	R/W	It decides the step value of delay line to increase during read leveling (default: 7'h1, fine step delay). It should be smaller than	7'b01

Field	Bit	R/W	Description	Initial Value
			7'b0F. [22:21]=2'b00 : The step value will be "rdlvl_incr_adj[20:16]" [22:21]=2'b01 : The step value will be "T/16" [22:21]=2'b10 : The step value will be "T/32" [22:21]=2'b11 : The step value will be "T/64"	
WrDeskew_clear	[14]	R/W	Clear WrDeSkewCode after Write Deskewing	1'b0
RdDeskew_clear	[13]	R/W	Clear RdDeSkewCode after Read Deskewing	1'b0
DLLDeskewEn	[12]	R/W	Deskew Code is updated with the latest Master DLL lock value whenever "dfi_ctrlupd_req" is issued from controller during DLLDeskewEn=1. It is required to compensate On-chip VT variation.	1'b0
rdlvl_start_adj	[11:8]	R/W	It decides the most left-shifted point when read leveling is started and the most right-shifted point when read leveling is ended. [9:8]=2'b00 : The most left-shifted code is 8'h00 [9:8]=2'b01 : The most left-shifted code is T/8 [9:8]=2'b10 : The most left-shifted code is T/8+T/16 [9:8]=2'b11 : The most left-shifted code is T/8-T/16 [11:10]=2'b00 : The most right-shifted Code is 8'hFF [11:10]=2'b01 : The most right-shifted Code is T/2+T/8 [11:10]=2'b10 : The most right-shifted Code is T/2+T/8+T/16 [11:10]=2'b11 : The most right-shifted Code is T/2+T/8-T/16	4'h0
InitDeskewEn	[6]	R/W	This field should be enabled before DQ Calibration is started.	1'b0
~~FastDeskewEn~~	~~[5]~~	~~R/W~~	~~Fast Deskew Enable signal. (It will be removed in final spec, Please don't use)~~	~~1'b0~~
~~FastDeskewStart~~	~~[4:2]~~	~~R/W~~	~~It controls the start code when Fast Deskew is enabled. It will decide "deskew start code" from "initial deskew code" depending on this setting. For exapme, if "initial deskew code" is 0x3F and FastDeskewStart=3'b001, "deskew start code" will be 0x3D (=0x3F-0x2). (It will be removed in final spec. Please don't use)~~ ~~3'b000 : T/3,~~ ~~3'b101 : T/4,~~ ~~3'b110 : T/5,~~ ~~3'b111 : T/6,~~ ~~Others : {FastDeskewStart[4:2], 1'b0}~~	~~3'h1~~
rdlvl_gateadj	[1:0]	R/W	It determines how much earlier ctrl_gate* is asserted than RDQS when the transition of RDQS is detected. [1:0]=2'b00 : T/2(default) [1:0]=2'b01 : T/4 [1:0]=2'b10 : T/8 [1:0]=2'b11 : T/16	2'h0

Table 4-5. PHY_CON5(Address Offset=0x14)

Field	Bit	R/W	Description	Initial Value

Field	Bit	R/W	Description	Initial Value
ReadModeCon	[7:0]	R/W	Read Register Mode Control After Gate Leveling, When ReadModeCon[7:0]='hD, Check "Gate Cycle"(0x50) When ReadModeCon[7:0]='hE, Check "Gate Code"(0x50) After Read Calibration, When ReadModeCon[7:0]='h05, Check "VWML"(0x50) When ReadModeCon[7:0]='h06, Check "VWMR"(0x50) When ReadModeCon[7:0]='h07, Check "VWMC"(0x50) When ReadModeCon[7:0]='h08, Check "VWMC"(0x50) When ReadModeCon[7:0]='h01, Check "Deskew Code"(0x50) After Write Calibration, When ReadModeCon[7:0]='h05, Check "VWML"(0x50) When ReadModeCon[7:0]='h06, Check "VWMR"(0x50) When ReadModeCon[7:0]='h07, Check "VWMC"(0x50) When ReadModeCon[7:0]='h09, Check "VWMC"(0x50) When ReadModeCon[7:0]='h03, Check "Deskew Code"(0x50) **NOTE:** The result of VWM* will be over-written after each calibration. Check PHY_CON18(=0x50) according to ReadModeCon[7:0].	0x0

According to ReadModeCon[7:0], You can get various status about PHY.

- When ReadModeCon[3:0]=4'h0, "Trddata_en" can be read from "PHY_CON18" register.

NOTE: For example, if ReadModeCon[7:0] = 8'h00, "Trddata_en" for each slice can be read from "PHY_CON18".

- According to ReadModeCon[7:0], "Read Deskew Code" can be read from PHY_CON19.
  - ReadModeCon[7:0]=8'h01 : DQ0 is selected.
  - ReadModeCon[7:0]=8'h11 : DQ1 is selected.
  - ReadModeCon[7:0]=8'h21 : DQ2 is selected.
  - ReadModeCon[7:0]=8'h31 : DQ3 is selected.
  - ReadModeCon[7:0]=8'h41 : DQ4 is selected.
  - ReadModeCon[7:0]=8'h51 : DQ5 is selected.
  - ReadModeCon[7:0]=8'h61 : DQ6 is selected.
  - ReadModeCon[7:0]=8'h71 : DQ7 is selected.

NOTE: For example, if ReadModeCon[7:0] = 8'h01, "Read Deskew Code" for DQ0 at each slice can be read from "PHY_CON19".

- According to ReadModeCon[7:0] and RdDeskew_clear, "ctrl_offsetr*"(=PHY_CON4) will be written to "Read Deskew Code" register instead of "ctrl_offsetr*" itself.
  - ReadModeCon[7:0]=8'h02 : DQ0 is written by "ctrl_offsetr*".
  - ReadModeCon[7:0]=8'h12 : DQ1 is written by "ctrl_offsetr*".
  - ReadModeCon[7:0]=8'h22 : DQ2 is written by "ctrl_offsetr*".
  - ReadModeCon[7:0]=8'h32 : DQ3 is written by "ctrl_offsetr*".
  - ReadModeCon[7:0]=8'h42 : DQ4 is written by "ctrl_offsetr*".
  - ReadModeCon[7:0]=8'h52 : DQ5 is written by "ctrl_offsetr*".
  - ReadModeCon[7:0]=8'h62 : DQ6 is written by "ctrl_offsetr*".

- ReadModeCon[7:0]=8'h72 : DQ7 is written by "ctrl_offsetr*".
- ReadModeCon[7:0]=8'h82 : All DQ Deskew Codes is cleared.

NOTE: After setting "ctrl_offsetr*" and ReadModeCon, assert and de-assert "RdDeskew_clear" to write the value of "ctrl_offsetr*".

- According to ReadModeCon[7:0], "Write Deskew Code" can be read from PHY_CON19.
  - ReadModeCon[7:0]=8'h03 : DQ0 is selected.
  - ReadModeCon[7:0]=8'h13 : DQ1 is selected.
  - ReadModeCon[7:0]=8'h23 : DQ2 is selected.
  - ReadModeCon[7:0]=8'h33 : DQ3 is selected.
  - ReadModeCon[7:0]=8'h43 : DQ4 is selected.
  - ReadModeCon[7:0]=8'h53 : DQ5 is selected.
  - ReadModeCon[7:0]=8'h63 : DQ6 is selected.
  - ReadModeCon[7:0]=8'h73 : DQ7 is selected.
  - ReadModeCon[7:0]=8'h83 : DM is selected.

NOTE: For example, if ReadModeCon[7:0] = 8'h03, "Write Deskew Code" for DQ0 at each slice can be read from "PHY_CON19".

- According to ReadModeCon[7:0] and WrDeskew_clear, "ctrl_offsetw*"(=PHY_CON6) will be written to "Write Deskew Code" register instead of "ctrl_offsetw*" itself.
  - ReadModeCon[7:0]=4'h04 : DQ0 is written by "ctrl_offsetw*".
  - ReadModeCon[7:0]=8'h14 : DQ1 is written by "ctrl_offsetw*".
  - ReadModeCon[7:0]=8'h24 : DQ2 is written by "ctrl_offsetw*".
  - ReadModeCon[7:0]=8'h34 : DQ3 is written by "ctrl_offsetw*".
  - ReadModeCon[7:0]=8'h44 : DQ4 is written by "ctrl_offsetw*".
  - ReadModeCon[7:0]=8'h54 : DQ5 is written by "ctrl_offsetw*".
  - ReadModeCon[7:0]=8'h64 : DQ6 is written by "ctrl_offsetw*".
  - ReadModeCon[7:0]=8'h74 : DQ7 is written by "ctrl_offsetw*".
  - ReadModeCon[7:0]=8'h84 : DM is written by "ctrl_offsetw*".
  - ReadModeCon[7:0]=8'h94 : All DQ Deskew Codes is cleared.

NOTE: After setting "ctrl_offsetw*" and ReadModeCon, assert and de-assert "WrDeskew_clear" to write the value of "ctrl_offsetw*".

- According to ReadModeCon[7:0], "Valid Window Margin Left" Register(=VWML) Code can be read from PHY_CON19.
  - ReadModeCon[7:0]=8'h05 : DQ0 is selected.
  - ReadModeCon[7:0]=8'h15 : DQ1 is selected.
  - ReadModeCon[7:0]=8'h25 : DQ2 is selected.
  - ReadModeCon[7:0]=8'h35 : DQ3 is selected.
  - ReadModeCon[7:0]=8'h45 : DQ4 is selected.
  - ReadModeCon[7:0]=8'h55 : DQ5 is selected.
  - ReadModeCon[7:0]=8'h65 : DQ6 is selected.
  - ReadModeCon[7:0]=8'h75 : DQ7 is selected.
  - ReadModeCon[7:0]=8'h85 : DM is selected.
  - ReadModeCon[7:0]=8'h95 : Write DQ VWML code is selected.

  - ReadModeCon[7:0]=8'hA5 : Write DM VWML code is selected.
  - ReadModeCon[7:0]=8'hB5 : Read DQ VWML code is selected.

NOTE: For example, if ReadModeCon[7:0] = 8'h15, "VWML" for DQ1 at each slice can be read from "PHY_CON19".

- According to ReadModeCon[7:0], "Valid Window Margin Right" Register(=VWMR) Code can be read from PHY_CON19.
  - ReadModeCon[7:0]=8'h06 : DQ0 is selected.
  - ReadModeCon[7:0]=8'h16 : DQ1 is selected.
  - ReadModeCon[7:0]=8'h26 : DQ2 is selected.
  - ReadModeCon[7:0]=8'h36 : DQ3 is selected.
  - ReadModeCon[7:0]=8'h46 : DQ4 is selected.
  - ReadModeCon[7:0]=8'h56 : DQ5 is selected.
  - ReadModeCon[7:0]=8'h66 : DQ6 is selected.
  - ReadModeCon[7:0]=8'h76 : DQ7 is selected.
  - ReadModeCon[7:0]=8'h86 : DM is selected.
  - ReadModeCon[7:0]=8'h96 : Write DQ VWMR code is selected.
  - ReadModeCon[7:0]=8'hA6 : Write DM VWMR code is selected.
  - ReadModeCon[7:0]=8'hB6 : Read DQ VWMR code is selected.

NOTE: For example, if ReadModeCon[7:0] = 8'h26, "VWMR" for DQ2 at each slice can be read from "PHY_CON19".

- According to ReadModeCon[7:0], "Valid Window Margin Center"(=VWMC) Code can be read from PHY_CON19.
  - ReadModeCon[7:0]=8'h07 : DQ0 is selected.
  - ReadModeCon[7:0]=8'h17 : DQ1 is selected.
  - ReadModeCon[7:0]=8'h27 : DQ2 is selected.
  - ReadModeCon[7:0]=8'h37 : DQ3 is selected.
  - ReadModeCon[7:0]=8'h47 : DQ4 is selected.
  - ReadModeCon[7:0]=8'h57 : DQ5 is selected.
  - ReadModeCon[7:0]=8'h67 : DQ6 is selected.
  - ReadModeCon[7:0]=8'h77 : DQ7 is selected.
  - ReadModeCon[7:0]=8'h87 : DM is selected.

NOTE: For example, if ReadModeCon[7:0] = 8'h17, "VWMC" for DQ1 at each slice can be read from "PHY_CON19".

- When ReadModeCon[3:0]=4'h8, "Read Valid Window Margin Center" Code can be read from PHY_CON19.
  - For example, if ReadModeCon[7:0] = 8'h08, "Read VWMC" for each slice can be read from "PHY_CON19".
- When ReadModeCon[3:0]=4'h9, "Write Valid Window Margin Center" Code is can be read from "PHY_CON19".
  - For example, if ReadModeCon[7:0] = 8'h09, "Write VWMC" for each slice can be read from "PHY_CON19".
- When ReadModeCon[3:0]=4'hA, "DM Valid Window Margin Center" Code can be read from "PHY_CON19".
  - For example, if ReadModeCon[7:0] = 8'h0A, "DM VWMC" for each slice can be read from "PHY_CON19".
- When ReadModeCon[3:0]=4'hB, "GATE Valid Window Margin Center" Code can be read from "PHY_CON19".
  - For example, if ReadModeCon[7:0] = 8'h0B, "GATE VWMC" for each slice can be read from

"PHY_CON19".

- When ReadModeCon[3:0]=4'hC, VWM Search Fail Status Register(=PHY_CON21) is selected. It should be read by zero if calibration is done normally.
  - For example, if ReadModeCon[7:0] = 8'h0C, " VWM Search Fail Status" for all DQs can be read from "PHY_CON19".
- When ReadModeCon[3:0]=4'hD, "GATE Centering Cycle Adjust" Register(=PHY_CON19) is selected. Controller can use the result of "GATE Leveling" by referencing these fields.
  - [2:0] : data_slice0 GATE Cycle Adjust value
  - [10:8] : data_slice1 GATE Cycle Adjust value
  - [18:16] : data_slice2 GATE Cycle Adjust value
  - [26:24] : data_slice3 GATE Cycle Adjust value

NOTE: For example, if ReadModeCon[7:0] = 8'h0D, "GATE Centering Cycle Adjust" for each slice can be read from "PHY_CON19".

- When ReadModeCon[3:0]=4'hE, "GATE Centering code" Register(=PHY_CON20) is selected at the initial stage. "GATE Centering code" will keep changing due to DLL update.
  - For example, if ReadModeCon[7:0] = 8'h0E, "GATE Centering code" for each slice can be read from "PHY_CON19".

Table 4-6. PHY_CON42(Address Offset=0xAC)

Field	Bit	R/W	Description	Initial Value
ctrl_bstlen	[12:8]	R/W	Burst Length(BL)	5'h0
ctrl_rdlat	[4:0]	R/W	Read Latency(RL)	5'h0

4.3 Read SDLL Code Control Register

Caution: Be careful that "ctrl_offsetr*" can be used for the other purpose(=Read Deskew Code Register). Please refer to ReadModeCon (=PHY_CON5).

Table 4-7. PHY_CON4(Address Offset=0x10)

Field	Bit	R/W	Description	Initial Value
ctrl_offsetr3(32bit)	[31:24]	R/W	This field can be used to give offset to read DQS. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. The right-shifted value is limited by the quarter of the maximum delay in Master Delay Line. Read DQS offset amount: ctrl_offsetr3[7] = 1: (tFS: fine step delay) Read DQS 90° delay amount – ctrl_offsetr0[6:0] x tFS ctrl_offsetr3[7] = 0: Read DQS 90° delay amount + ctrl_offsetr0[6:0] x tFS	0x8
ctrl_offsetr2(32bit)	[23:16]	R/W	This field can be used to give offset to read DQS.	0x8

Field	Bit	R/W	Description	Initial Value
			If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. The right-shifted value is limited by the quarter of the maximum delay in Master Delay Line. Read DQS offset amount: ctrl_offsetr2[7] = 1: (tFS: fine step delay) Read DQS 90° delay amount – ctrl_offsetr0[6:0] x tFS ctrl_offsetr2[7] = 0: Read DQS 90° delay amount + ctrl_offsetr0[6:0] x tFS	
ctrl_offsetr1	[15:8]	R/W	This field can be used to give offset to read DQS. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. The right-shifted value is limited by the quarter of the maximum delay in Master Delay Line. Read DQS offset amount: ctrl_offsetr1[7] = 1: (tFS: fine step delay) Read DQS 90° delay amount – ctrl_offsetr0[6:0] x tFS ctrl_offsetr1[7] = 0: Read DQS 90° delay amount + ctrl_offsetr0[6:0] x tFS	0x8
ctrl_offsetr0	[7:0]	R/W	This field can be used to give offset to read DQS. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. The right-shifted value is limited by the quarter of the maximum delay in Master Delay Line. Read DQS offset amount: ctrl_offsetr0[7] = 1: (tFS: fine step delay) Read DQS 90° delay amount – ctrl_offsetr0[6:0] x tFS ctrl_offsetr0[7] = 0: Read DQS 90° delay amount + ctrl_offsetr0[6:0] x tFS	0x8

To capture the DQs with DQS in read operation, DQS is shifted by 90° and this makes rising edge of DQS be located in the center of valid window of DQs. But according to the channel condition, i.e. PCB or SSN, rising edge of DQS could be biased from the center of valid window of DQs. ctrl_offset* is used to compensate the biased DQS and make it be placed in the center of DQs. If ctrl_offset*[6] is 0, ctrl_offset*[5:0] x tFS (tFS: fine step delay) is added to the 90° delay amount and if ctrl_offset*[6] is 1, ctrl_offset*[5:0] x tFS is subtracted from the 90° delay amount. ctrl_offset*[5:0] means the number of delay cells of the "delay line". Generally, ctrl_offset*[6:0] would be zero.

dfi_ctrlupd_req signal is required to update delay line control signals before the first normal operation after checking whether dfi_init_complete is set and at every memory refresh cycle. Before set and clear this signal during initialization, dfi_init_complete signal should be checked whether it's HIGH to confirm that DLL is locked.

4.4 Write SDLL Code Control Register

Caution: Be careful that "ctrl_offsetw*" can be used for the other purpose (=Write Deskew Code Register). Please refer to ReadModeCon (=PHY_CON5).

Table 4-8. PHY_CON6(Address Offset=0x18)

Field	Bit	R/W	Description	Initial Value
ctrl_offsetw3	[31:24]	R/W	This field can be used to give offset to write DQ. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. The right-shifted value is limited by the quarter of the maximum delay in Master Delay Line. Write DQ offset amount : ctrl_offsetw2[7] = 1 : (tFS: fine step delay) Write DQ 270° delay amount – ctrl_offsetw2[6:0] x tFS ctrl_offsetw2[7] = 0 : Write DQ 270° delay amount + ctrl_offsetw2[6:0] x tFS	0x8
ctrl_offsetw2	[23:16]	R/W	This field can be used to give offset to write DQ. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. The right-shifted value is limited by the quarter of the maximum delay in Master Delay Line. Write DQ offset amount : ctrl_offsetw2[7] = 1 : (tFS: fine step delay) Write DQ 270° delay amount – ctrl_offsetw2[6:0] x tFS ctrl_offsetw2[7] = 0 : Write DQ 270° delay amount + ctrl_offsetw2[6:0] x tFS	0x8
ctrl_offsetw1	[15:8]	R/W	This field can be used to give offset to write DQ. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. The right-shifted value is limited by the quarter of the maximum delay in Master Delay Line. Write DQ offset amount : ctrl_offsetw2[7] = 1 : (tFS: fine step delay) Write DQ 270° delay amount – ctrl_offsetw2[6:0] x tFS ctrl_offsetw2[7] = 0 : Write DQ 270° delay amount + ctrl_offsetw2[6:0] x tFS	0x8
ctrl_offsetw0	[7:0]	R/W	This field can be used to give offset to write DQ. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. The right-shifted value is limited by the quarter of the maximum delay in Master Delay Line. Write DQ offset amount : ctrl_offsetw2[7] = 1 : (tFS: fine step delay)	0x8

Field	Bit	R/W	Description	Initial Value
			Write DQ 270° delay amount – ctrl_offsetw2[6:0] x tFS ctrl_offsetw2[7] = 0 : Write DQ 270° delay amount + ctrl_offsetw2[6:0] x tFS	

Table 4-9. PHY_CON7(Address Offset=0x1C)

Field	Bit	R/W	Description	Initial Value
~~ctrl_offsetw4 (40bit)~~	~~[7:0]~~	~~R/W~~	~~This field can be used to give offset to write DQ.~~ ~~If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. The right-shifted value is limited by the quarter of the maximum delay in Master Delay Line.~~ ~~Write DQ offset amount :~~ ~~ctrl_offsetw2[7] = 1 : (tFS: fine step delay)~~ ~~Write DQ 270° delay amount – ctrl_offsetw2[5:0] x tFS~~ ~~ctrl_offsetw2[7] = 0 :~~ ~~Write DQ 270° delay amount + ctrl_offsetw2[5:0] x tFS~~	~~0x8~~

4.5 GATE SDLL Code Control Register

Table 4-10. PHY_CON3(Address Offset=0x0C)

Field	Bit	R/W	Description	Initial Value
~~ctrl_shiftc4(40bit)~~	~~[22:20]~~	~~R/W~~	~~GATEin signal delay amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. This value is limited by the half of the maximum delay in Master Delay Line. GATEin signal will be delayed by T/16 whenever it increases one.~~ ~~000: 0(0° shift), 001: T(365° shift),~~ ~~010: T/2(180° shift), 011: T/4(90° shift)~~ ~~100: T/8(45° shift), 101: T/16(22.5° shift)~~	~~3'b010~~
ctrl_shiftc3(32bit)	[17:15]	R/W	GATEin signal delay amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. This value is limited by the half of the maximum delay in Master Delay Line. GATEin signal will be delayed by T/16 whenever it increases one. 000: 0(0° shift), 001: T(365° shift), 010: T/2(180° shift), 011: T/4(90° shift) 100: T/8(45° shift), 101: T/16(22.5° shift)	3'b010
ctrl_shiftc2(32bit)	[12:10]	R/W	GATEin signal delay amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. This value is	3'b010

Field	Bit	R/W	Description	Initial Value
			limited by the half of the maximum delay in Master Delay Line. GATEin signal will be delayed by T/16 whenever it increases one. 000: 0(0° shift), 001: T(365° shift), 010: T/2(180° shift), 011: T/4(90° shift) 100: T/8(45° shift), 101: T/16(22.5° shift)	
ctrl_shiftc1	[7:5]	R/W	GATEin signal delay amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. This value is limited by the half of the maximum delay in Master Delay Line. GATEin signal will be delayed by T/16 whenever it increases one. 000: 0(0° shift), 001: T(365° shift), 010: T/2(180° shift), 011: T/4(90° shift) 100: T/8(45° shift), 101: T/16(22.5° shift)	3'b010
ctrl_shiftc0	[2:0]	R/W	GATEin signal delay amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. This value is limited by the half of the maximum delay in Master Delay Line. GATEin signal will be delayed by T/16 whenever it increases one. 000: 0(0° shift), 001: T(365° shift), 010: T/2(180° shift), 011: T/4(90° shift) 100: T/8(45° shift), 101: T/16(22.5° shift)	3'b010

Table 4-11. PHY_CON8(Address Offset=0x20)

Field	Bit	R/W	Description	Initial Value
ctrl_offsetc3	[31:24]	R/W	Gate offset amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. ctrl_offsetc [7] = 1 : (tFS: fine step delay) GATEout delay amount – ctrl_offsetc [6:0] x tFS ctrl_offsetc [7] = 0 : GATEout delay amount + ctrl_offsetc [6:0] x tFS	0x0
ctrl_offsetc2	[23:16]	R/W	Gate offset amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. ctrl_offsetc [7] = 1 : (tFS: fine step delay) GATEout delay amount – ctrl_offsetc [6:0] x tFS ctrl_offsetc [7] = 0 : GATEout delay amount + ctrl_offsetc [6:0] x tFS	0x0
ctrl_offsetc1	[15:8]	R/W	Gate offset amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW.	0x0

Field	Bit	R/W	Description	Initial Value
			ctrl_offsetc [7] = 1 : (tFS: fine step delay) GATEout delay amount – ctrl_offsetc [6:0] x tFS ctrl_offsetc [7] = 0 : GATEout delay amount + ctrl_offsetc [6:0] x tFS	
ctrl_offsetc0	[7:0]	R/W	Gate offset amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. ctrl_offsetc [7] = 1 : (tFS: fine step delay) GATEout delay amount – ctrl_offsetc [6:0] x tFS ctrl_offsetc [7] = 0 : GATEout delay amount + ctrl_offsetc [6:0] x tFS	0x0

Table 4-12. PHY_CON9(Address Offset=0x24)

Field	Bit	R/W	Description	Initial Value
Reserved	[31:8]		Should be zero	0x0
ctrl_offsetc4 (40bit)	[7:0]	R/W	Gate offset amount for DDR. If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. ctrl_offsetc [7] = 1 : (tFS: fine step delay) GATEout delay amount – ctrl_offsetc [6:0] x tFS ctrl_offsetc [7] = 0 : GATEout delay amount + ctrl_offsetc [6:0] x tFS	0x0

4.6 Command SDLL Code Control Register

Table 4-13. PHY_CON10(Address Offset=0x28)

Field	Bit	R/W	Description	Initial Value
Reserved	[31:25]		Should be zero	0x0
ctrl_resync	[24]	R/W	Active RISIG-EDGE signal. This signal should become LOW after set HIGH for normal operation. When this bit transits from LOW to HIGH, pointers of FIFO and DLL information is updated. In general, this bit should be set during initialization and refresh cycles. Refer to "MC-Initiated Update"(p105) to use ctrl_resync.	0x0
Reserved	[23:8]		Should be zero	0x0
ctrl_offsetd	[7:0]	R/W	This field is for debug purpose. (For LPDDR2) If this field is fixed, this should not be changed during operation. This value is valid only after dfi_ctrlupd_req becomes HIGH and LOW. The right-shifted value is limited by the quarter of the maximum delay in Master Delay Line. offset amount for 270° clock generation:	0x8

Field	Bit	R/W	Description	Initial Value

			ctrl_offsetd[7] = 1 : (tFS: fine step delay)
			270° delay amount – ctrl_offsetd[6:0] x tFS
			ctrl_offsetd[7] = 0 :
			270° delay amount + ctrl_offsetd[6:0] x tFS	

4.7 MDLL Control Register

Table 4-14. PHY_CON12(Address Offset=0x30)

Field	Bit	R/W	Description	Initial Value
Reserved	[31]		Should be zero	0x0
ctrl_start_point	[30:24]	R/W	Initial DLL lock start point. This is the number of delay cells and is the start point where "DLL" start tracing to be locked. Initial delay time is calculated by multiplying the unit delay of delay cell and this value.	7'h10
Reserved	[23]		Should be zero	0x0
ctrl_inc	[22:16]	R/W	Increase amount of start point	7'h10
ctrl_force	[14:8]	R/W	This field is used instead of ctrl_lock_value[9:2] found by the DLL only when ctrl_dll_on is LOW ,i.e. If the DLL is off, this field is used to generate 270° clock and shift DQS by 90°.	0x0
ctrl_start	[6]	R/W	This field is used to start DLL locking.	0'b1
ctrl_dll_on	[5]	R/W	HIGH active start signal to turn on the DLL. This signal should be kept HIGH for normal operation. If this signal becomes LOW, DLL is turned off. This bit should be kept set before ctrl_start is set to turn on the DLL.	0'b1
ctrl_ref	[4:1]	R/W	This field determines the period of time when ctrl_locked is cleared. 4’b0000: Don’t use. 4’b0001: ctrl_flock is de-asserted during 16 clock cycles, ctrl_locked is deasserted. 4’b0010: ctrl_flock is de-asserted during 24 clock cycles, ctrl_locked is deasserted. ~ 4’b1110: ctrl_flock is de-asserted during 120 clock cycles, ctrl_locked is deasserted. 4’b1111: Once ctrl_locked and dfi_init_complete are asserted, those won’t be deasserted until rst_n is asserted.	4'h8
Reserved	[0]		Should be zero	0x0

Table 4-15. PHY_CON13(Address Offset=0x34)

Field	Bit	R/W	Description	Initial Value

Field	Bit	R/W	Description	Initial Value
ctrl_lock_value	[16:8]	R/W	Locked delay line encoding value. ctrl_lock_value[8:2] : number of delay cells for coarse lock. ctrl_lock_value[1:0] : control value for fine lock. From ctrl_lock_value[8:0], tFS(fine step delay) can be calculated. tFS = tCK / ctrl_lock_value[9:0].	-
ctrl_clock	[2]	R	Coarse lock information. According to clock jitter, ctrl_clock can be de-asserted.	-
ctrl_flock	[1]	R	Fine lock information. According to clock jitter, ctrl_flock can be de-asserted.	-
ctrl_locked	[0]	R	DLL stable lock information. This field is set after ctrl_flock is set. This field is cleared when ctrl_flock is de-asserted for some period of time specified by ctrl_ref[3:0] value. This field is required for stable lock status check.	-

"DLL" is used to get how many delay cells should be passed through the "delay line" to delay a signal to an amount of one clock period and is controlled by ctrl_start, ctrl_start_point and ctrl_inc.

ctrl_start should be set and kept high to make "DLL" keep tracing one clock period after clock(PHY clock) becomes stable. If ctrl_start becomes LOW, "DLL" stops tracing one clock period.

ctrl_clock and ctrl_flock are status fields indicating whether "DLL" is locked. After ctrl_start becomes HIGH, DLL starts tracing one clock period and controls (increases or decreases) the number of delay cells for the clock to pass through. And if ctrl_clock is set("DLL" is locked), "DLL" changes step delays of the "delay line" and controls the "delay line" in fine resolution to reduce the "phase offset error". When ctrl_flock is set, "DLL" is locked with fine resolution. ctrl_lock_value is information field to indicate the number of delay cells to delay a signal to one clock period through the "delay line".

- {ctrl_clock, ctrl_flock = 2'b00} : DLL is not locked.
- {ctrl_clock, ctrl_flock = 2'b01} : Impossible value.
- {ctrl_clock, ctrl_flock = 2'b10} : Locked.
- {ctrl_clock, ctrl_flock = 2'b11} : Locked.

NOTE: * Recommended value for ctrl_start_point[7:0] and ctrl_inc[7:0] are 8'h10.

4.8 Low Power Control Register

Table 4-16. PHY_CON14(Address Offset=0x38)

Field	Bit	R/W	Description	Initial Value
ctrl_pulld_dq	[11:8]	R/W	Active HIGH signal to down DQ signals. For normal operation this field should be zero. When bus is idle, it can be set to pull-down or up the bus not to make bus high-z state.	4'h0
ctrl_pulld_dqs	[3:0]	R/W	Active HIGH signal to pull-up or down PDQS/NDQS signals.	4'h0

Field	Bit	R/W	Description	Initial Value
			When using Gate Leveling in DDR3, this field can be zero. When bus is idle, it can be set to pull-down or up the bus not to make bus high-z state (PDQS/NDQS is pulled-down/up). For LPDDR2/LPDDR3 mode, this field should be always set for normal operation to make P/NDQS signals pull-down/up.	

4.9 Feedback Control Register

Table 4-17. PHY_CON15(Address Offset=0x3C)

Field	Bit	R/W	Description	Initial Value
~~ctrl_fb_err~~	~~[20:16]~~	~~R~~	~~Feedback test stop with error for each slice.~~ ~~Ctrl_fb_err[0] : error of data channel0.~~ ~~Ctrl_fb_err[1] : error of data channel1.~~ ~~Ctrl_fb_err[2] : error of data channel2 for 32-bit PHY.~~ ~~Error of control for 16-bit PHY.~~ ~~Ctrl_fb_err[3] : error of data channel3 for 32-bit PHY.~~ ~~Ctrl_fb_err[4] : error of control channel for 32-bit PHY.~~	~~5'h0~~
~~ctrl_fb_okay~~	~~[12:8]~~	~~R~~	~~Feedback test completion without error for each slice.~~ ~~Ctrl_fb_oky[0] : okay of data channel0.~~ ~~Ctrl_fb_oky[1] : okay of data channel1.~~ ~~Ctrl_fb_oky[2] : okay of data channel2 for 32-bit PHY.~~ ~~Okay of control channel for 16-bit PHY.~~ ~~Ctrl_fb_oky[3] : okay of data channel3 for 32-bit PHY.~~ ~~Ctrl_fb_oky[4] : okay of control channel for 32-bit PHY.~~	~~5'h0~~
~~ctrl_fb_start~~	~~[4:0]~~	~~R/W~~	~~Feedback test start signal for each slice.~~ ~~Ctrl_fb_start[0]: start of data channel0~~ ~~ctrl_fb_start[1]: start of data channel1~~ ~~ctrl_fb_start[2]: start of data channel2 for 32-bit PHY~~ ~~start of control channel for 16-bit PHY~~ ~~ctrl_fb_start[3]: start of data channel3 for 32-bit PHY.~~ ~~Ctrl_fb_start[4]: start of control channel for 32-bit PHY.~~ ~~For this test, mode_highz should be set when memory is on the board~~	~~5'h0~~

4.10 ZQ Calibration Control Register

Table 4-18. PHY_CON16(Address Offset=0x40)

Field	Bit	R/W	Description	Initial Value
Reserved	[31:28]		Should be zero	0x0
zq_clk_en	[27]	R/W	ZQ I/O Clock enable	1'b1

Field	Bit	R/W	Description	Initial Value
zq_mode_dds	[26:24]	R/W	Driver strength selection. . It recommends one of the following settings instead of 3'h0. 3'b100 : 48Ω Impedance output driver 3'b101 : 40Ω Impedance output driver 3'b110 : 34Ω Impedance output driver 3'b111 : 30Ω Impedance output driver	3'h0
zq_mode_term	[23:21]	R/W	On-die-termination(ODT) resistor value selection. "pblpddr3_dds" and "pblpddr3_dqs_dds" don't support ODT. 3'b001 : 120Ω Receiver termination 3'b010 : 60Ω Receiver termination 3'b011 : 40Ω Receiver termination 3'b100 : 30Ω Receiver termination	3'h0
zq_rgddr3	[20]	R/W	GDDR3 mode enable signal(High: GDDR3 mode)	1'b0
zq_mode_noterm	[19]	R/W	Termination disable selection. 1 : termination disable. 0 : termination enable. DDR : 1'b1 DDR2/DDR3 : 1'b0(recommended) or 1'b1(when termination is not used) LPDDR2/LPDDR3 : 1'b1 gDDR3 : 1'b0	1'b0
zq_clk_div_en	[18]	R/W	Clock dividing enable	1'b0
zq_force_impn	[17:15]	R/W	Immediate control code for pull-down.	3'h0
zq_force_impp	[14:12]	R/W	Immediate control code for pull-up.	3'h0
zq_udt_dly	[11:4]	R/W	ZQ I/O clock enable duration for auto calibration mode.	8'h30
zq_manual_mode	[3:2]	R/W	Manual calibration mode selection 2'b00: force calibration 2'b01: long calibration 2;b10: short calibration	2'b01
zq_manual_str	[1]	R/W	Manual calibration start	1'b0
zq_auto_en	[0]	R/W	Auto calibration enable	1'b0

NOTE: "zq_manual_str"(=PHY_CON16[1]) should be toggled after PHY_CON16[17:2] or PHY_CON[26:19] is changed. For example, if zq_mode_dds is written by 3'b111, "zq_manual_str" should be set and cleared to apply this new strength value(3'b111).

The customer should find out the optimal value for ZQ I/O interface during the real application test, because the optimal value for ZQ I/O interface can be changed depending on the real application.

Long calibration and short calibration modes are supported for ZQ I/O calibration. With ZQ I/O calibration driving impedance and termination impedance can be calibrated with RZQ. ZQ I/O calibrates the I/Os to match the driving and termination impedance by referencing resistor value of RZQ.

Table 4-19. PHY_CON17(Address Offset=0x48)

Field	Bit	R/W	Description	Initial Value
Reserved	[31:9]		Should be zero	-
zq_pmon	[8:6]	R	Control code found by auto calibration for pull-up.	-
zq_nmon	[5:3]	R	Control code found by auto calibration for pull-down.	-
zq_error	[2]	R	Calibration fail indication (High: calibration failed)	-
zq_pending	[1]	R	Auto calibration enable status	-
zq_done	[0]	R	ZQ Calibration is finished.	1'b0

NOTE: Refer to ZQ I/O calibration section in 8 application note.

Table 4-20. PHY_CON40(Address Offset=0xA4)

Field	Bit	R/W	Description	Initial Value
ctrl_zq_clk_div	[31:0]	R/W	ZQ Clock divider setting value	0x7

Table 4-21. PHY_CON41(Address Offset=0xA8)

Field	Bit	R/W	Description	Initial Value
ctrl_zq_timer	[31:0]	R/W	It controls the interval between each ZQ calibration	0xF0

4.11 Read Data Enable Timing Status Register

Table 4-22. PHY_CON18(Address Offset=0x4C)

Field	Bit	R/W	Description	Initial Value
dm_fail_status	[27:24]	R	It will be enabled if there is no pass period after DM Calibration. It should be read by zero if Calibration is done normally. Please refer to ReadModeCon(=PHY_CON5) to read.	0x0
T3_rddata_en	[23:18]	R	Trddata_en timing parameter for data slice 3. Please refer to ReadModeCon(=PHY_CON5) to read.	0x15
T2_rddata_en	[17:12]	R	Trddata_en timing parameter for data slice 2. Please refer to ReadModeCon(=PHY_CON5) to read.	0x15
T1_rddata_en	[11:6]	R	Trddata_en timing parameter for data slice 1. Please refer to ReadModeCon(=PHY_CON5) to read.	0x15
T0_rddata_en	[5:0]	R	Trddata_en timing parameter for data slice 0. Please refer to ReadModeCon(=PHY_CON5) to read.	0x15

4.12 Read Leveling Status Register

Table 4-23. PHY_CON19(Address Offset=0x50)

Field	Bit	R/W	Description	Initial Value
rdlvl_offsetr3	[31:24]	R	DQ Calibration SDLL code for data slice 3. Please refer to ReadModeCon(=PHY_CON5) to read.	-
rdlvl_offsetr2	[23:16]	R	DQ Calibration SDLL code for data slice 2. Please refer to ReadModeCon(=PHY_CON5) to read.	-
rdlvl_offsetr1	[15:8]	R	DQ Calibration SDLL code for data slice 1. Please refer to ReadModeCon(=PHY_CON5) to read.	-
rdlvl_offsetr0	[7:0]	R	DQ Calibration SDLL code for data slice 0. Please refer to ReadModeCon(=PHY_CON5) to read.	-

Table 4-24. PHY_CON20(Address Offset=0x54)

Field	Bit	R/W	Description	Initial Value
rdlvl_offsetc3	[31:24]	R	Gate Training SDLL code for data slice 3. Please refer to ReadModeCon(=PHY_CON5) to read.	-
rdlvl_offsetc2	[23:16]	R	Gate Training SDLL code for data slice 2. Please refer to ReadModeCon(=PHY_CON5) to read.	-
rdlvl_offsetc1	[15:8]	R	Gate Training SDLL code for data slice 1. Please refer to ReadModeCon(=PHY_CON5) to read.	-
rdlvl_offsetc0	[7:0]	R	Gate Training SDLL code for data slice 0. Please refer to ReadModeCon(=PHY_CON5) to read.	-

Table 4-25. PHY_CON21(Address Offset=0x58)

Field	Bit	R/W	Description	Initial Value
vwm_fail_status	[31:0]	R	It will be enabled if there is no pass period after DQ Calibration. It should be read by zero if Calibration is done normally. Please refer to ReadModeCon(=PHY_CON5) to read.	0x0

4.13 Read Leveling Control Register

Table 4-26. PHY_CON22(Address Offset=0x5C)

Field	Bit	R/W	Description	Initial Value
lpddr2_addr	[19:0]	R/W	LPDDR2/LPDDR3 Address. Default value(=0x208) is Mode Register Reads to DQ Calibration registers MR32. Reads to MR32 return DQ Calibration Pattern "1111-0000-1111-0000" on DQ[0], DQ[8], DQ[16], and DQ[24] for x32 devices. When doing Write Training, This field should be set by READ command. For	0x208

Field	Bit	R/W	Description	Initial Value
			example, lpddr2_addr2 will be 20'h5 if the column address is 11'h0 and bank address is 3'b000. According to READ Command definition in LPDDR2 or LPDDR3 lpddr2_addr[19:0] = "C11-C10-C9-C8-C7-C6-C5-C4-C3-AP-BA2-BA1-BA0-C2-C1-R-R-H-L-H" (C means Column Address, BA means Bank Address, R means Reserved) In case of CA swap mode, lpddr2_addr=20'h41 for Read Training and lpddr2_addr=20'h204 for Write Training if the column address is 11'h0 and bank address is 3'b000. lpddr2_addr[19:0] = "AP-C3-C9-C5-C6-C7-C8-C4-C10-C11-H-L-BA0-R-R-C1-C2-H-B1-B2"(CA swap mode)	

Table 4-27. PHY_CON23(Address Offset=0x60)

Field	Bit	R/W	Description	Initial Value
lpddr2_default	[19:0]	R/W	LPDDR2/LPDDR3 Default Address	0x03FF

Table 4-28. PHY_CON24(Address Offset=0x64)

Field	Bit	R/W	Description	Initial Value
ddr3_default	[31:16]	R/W	DDR3 Default Address	0x0
ddr3_addr	[15:0]	R/W	DDR3 Address	0x0
ca_swap_mode	[0]	R/W	If ctrl_ddr_mode[1]=1 and ca_swap_mode=1, PHY will be in "CA swap mode" for POP. In "CA swap mode", CA[9:0] will be swapped in the following way. CA[0] → CA[9] CA[1] → CA[8] CA[2] → CA[7] CA[3] → CA[6] CA[4] → CA[5] CA[5] → CA[4] CA[6] → CA[3] CA[7] → CA[2] CA[8] → CA[1] CA[9] → CA[0] **NOTE:** Don't use "ctrl_atgate=1" in normal operation when ca_swap_mode = 1. "ctrl_atgate" can be enabled only during calibration.	0x0

Table 4-29. PHY_CON25(Address Offset=0x68)

Field	Bit	R/W	Description	Initial Value

Field	Bit	R/W	Description	Initial Value
ddr3_cmd	[29:16]	R/W	DDR3 Command	0x105E
lpddr2_cmd	[13:0]	R/W	LPDDR2/3 Command. This field should be "16'h000E" using LPDDR2 or LPDDR3.	0x107E

NOTE: Please refer to the meaning of each bit field in "cmd_default", "ddr3_cmd" and lpddr2_cmd.
[1:0] : CS[1:0]
[2:3] : CKE[1:0]
[4] : WEN,
[5] : CAS
[6] : RAS
[8:7] : ODT[1:0] (Not applicable)
[11:9] : BANK[2:0]
[12] : RESET

Table 4-30. PHY_CON26(Address Offset=0x6C)

Field	Bit	R/W	Description	Initial Value
T_wrdata_en	[20:16]	R/W	Clock cycles between write command and the first edge of DQS which can capture the first valid DQ. For example, if WL is 6, It should be set as 7(=WL+1) in LPDDR3, 6(=WL) in DDR3.	0x8
cmd_default	[13:0]	R/W	Default Command 16'h000F(LPDDR2, LPDDR3) 16'h107F(DDR2, DDR3)	0x107F

4.14 Valid Window Margin Register

Table 4-31. PHY_CON27(Address Offset=0x70)

Field	Bit	R/W	Description	Initial Value
rlvl_vwml3	[31:24]	R	Left Code Value in Read Valid Window Margin for Data Slice3. Please refer to ReadModeCon(=PHY_CON5) to read.	-
rlvl_vwml2	[23:16]	R	Left Code Value in Read Valid Window Margin for Data Slice2. Please refer to ReadModeCon(=PHY_CON5) to read.	-
rlvl_vwml1	[15:8]	R	Left Code Value in Read Valid Window Margin for Data Slice1. Please refer to ReadModeCon(=PHY_CON5) to read.	-
rlvl_vwml0	[7:0]	R	Left Code Value in Read Valid Window Margin for Data Slice0. Please refer to ReadModeCon(=PHY_CON5) to read.	-

Table 4-32. PHY_CON28(Address Offset=0x74)

Field	Bit	R/W	Description	Initial Value
rlvl_vwmr3	[31:24]	R	Right Code Value in Read Valid Window Margin for Data Slice3.	-

Field	Bit	R/W	Description	Initial Value
			Please refer to ReadModeCon(=PHY_CON5) to read.	
rlvl_vwmr2	[23:16]	R	Right Code Value in Read Valid Window Margin for Data Slice2. Please refer to ReadModeCon(=PHY_CON5) to read.	-
rlvl_vwmr1	[15:8]	R	Right Code Value in Read Valid Window Margin for Data Slice1. Please refer to ReadModeCon(=PHY_CON5) to read.	-
rlvl_vwmr0	[7:0]	R	Right Code Value in Read Valid Window Margin for Data Slice0. Please refer to ReadModeCon(=PHY_CON5) to read.	-

4.15 Version Information Register

Table 4-33. PHY_CON29(Address Offset=0x78)

Field	Bit	R/W	Description	Initial Value
Version_Info	[31:0]	R	Version Information	0x0501_0203

4.16 Write Leveling Control Register

Table 4-34. PHY_CON30(Address Offset=0x7C)

Field	Bit	R/W	Description	Initial Value
ctrl_wrlvl3_code	[30:24]	R/W	Write Level Slave DLL Code Value for Data_Slice 3(0x8~0x38)	0x0
ctrl_wrlvl2_code	[23:17]	R/W	Write Level Slave DLL Code Value for Data_Slice 2(0x8~0x38)	0x0
ctrl_wrlvl	[16]	R/W	Write Level Enable	0x0
ctrl_wrlvl1_code	[14:8]	R/W	Write Level Slave DLL Code Value for Data_Slice 1(0x8~0x38)	0x0
ctrl_wrlvl0_code	[6:0]	R/W	Write Level Slave DLL Code Value for Data_Slice 0(0x8~0x38)	0x0

4.17 DeSkew Control Register

Table 4-35. PHY_CON31(Address Offset=0x80)

Field	Bit	R/W	Description	Initial Value
CA4DeSkewCode	[31:28]	R/W	DeSkew Code for CA[4] (0x8~0x60)	0x0
CA3DeSkewCode	[27:21]	R/W	DeSkew Code for CA[3] (0x8~0x60)	0x0
CA2DeSkewCode	[20:14]	R/W	DeSkew Code for CA[2] (0x8~0x60)	0x0
CA1DeSkewCode	[13:7]	R/W	DeSkew Code for CA[1] (0x8~0x60)	0x0
CA0DeSkewCode	[6:0]	R/W	DeSkew Code for CA[0] (0x8~0x60)	0x0

Table 4-36. PHY_CON32(Address Offset=0x84)

Field	Bit	R/W	Description	Initial Value
CA9DeSkewCode	[31]	R/W	DeSkew Code for CA[9] (0x8~0x60)	0x0
CA8DeSkewCode	[30:24]	R/W	DeSkew Code for CA[8] (0x8~0x60)	0x0
CA7DeSkewCode	[23:17]	R/W	DeSkew Code for CA[7] (0x8~0x60)	0x0
CA6DeSkewCode	[16:10]	R/W	DeSkew Code for CA[6] (0x8~0x60)	0x0
CA5DeSkewCode	[9:3]	R/W	DeSkew Code for CA[5] (0x8~0x60)	0x0
CA4DeSkewCode	[2:0]	R/W	DeSkew Code for CA[4] (0x8~0x60)	0x0

NOTE: When reading, The value of "PHY_CON32" should be read from "{PHY_CON33[1], PHY_CON32[31:1]}".

Table 4-37. PHY_CON33(Address Offset=0x88)

Field	Bit	R/W	Description	Initial Value
CKE0DeSkewCode	[31:27]	R/W	DeSkew Code for CKE[0], (0x8~0x60)	0x0
CS1DeSkewCode	[26:20]	R/W	DeSkew Code for CS[1] (0x8~0x60)	0x0
CS0DeSkewCode	[19:13]	R/W	DeSkew Code for CS[0] (0x8~0x60)	0x0
CKDeSkewCode	[12:6]	R/W	DeSkew Code for CK, (0x8~0x60)	0x0
CA9DeSkewCode	[5:0]	R/W	DeSkew Code for CA[9] (0x8~0x60)	0x0

NOTE: When reading, The value of "PHY_CON33" should be read from "{PHY_CON34[2:1], PHY_CON33[31:2]}".

Table 4-38. PHY_CON34(Address Offset=0x8C)

Field	Bit	R/W	Description	Initial Value
CKE1DeSkewCode	[8:2]	R/W	DeSkew Code for CKE[1] (0x8~0x60)	0x0
CKE0DeSkewCode	[1:0]	R/W	DeSkew Code for CKE[0] (0x8~0x60)	0x0

NOTE: When reading, The value of "PHY_CON34[8:0]" should be read from "{ PHY_CON34[0],PHY_CON34[10:3]}".

Caution: If the DeSkew Code for CS[1:0], CK is changed, CKE should be always "LOW" during updating. If the DeSkew Code for CKE[1:0] is changed, Please initialize memory again.

Table 4-39. PHY_CON37(Address Offset=0x98)

Field	Bit	R/W	Description	Initial Value
RSTDeSkewCode	[5:0]	R/W	DeSkew Code for Reset (0x8~0x38)	0x0

Table 4-40. PHY_CON39(Address Offset=0xA0)

Field	Bit	R/W	Description	Initial Value
Da3DS	[27:25]	R/W	Driver Strength Selection for Data Slice 3	0x0

Field	Bit	R/W	Description	Initial Value
Da2DS	[24:22]	R/W	Driver Strength Selection for Data Slice 2	0x0
Da1DS	[21:19]	R/W	Driver Strength Selection for Data Slice 1	0x0
Da0DS	[18:16]	R/W	Driver Strength Selection for Data Slice 0	0x0
CaCkDrvrDS	[11:9]	R/W	Driver Strength Selection for CK	0x0
CaCkeDrvrDS	[8:6]	R/W	Driver Strength Selection for Cke[1:0]	0x0
CaCSDrvrDS	[5:3]	R/W	Driver Strength Selection for CS[1:0]	0x0
CaAdrDrvrDS	[2:0]	R/W	Driver Strength Selection for ADCT[15:0].	0x0

NOTE: It recommends that Driver Strength will be one of the following settings instead of 3'h0.
3'b100 : 48Ω Impedance output driver
3'b101 : 40Ω Impedance output driver
3'b110 : 34Ω Impedance output driver
3'b111 : 30Ω Impedance output driver

Caution: When selecting "pblpddr3_dds" and "pblpddr3_dqs_dds"(Refer to User Guide 2. I/O SELECTION), "PHY_CON39" can control Driver Strength. If using "pblpddr3" and "pblpddr3_dqs" instead, "zq_mode_dds"(=PHY_CON16[26:24]) will control Driver Strength.

5 Functional Description

5.1 DLL

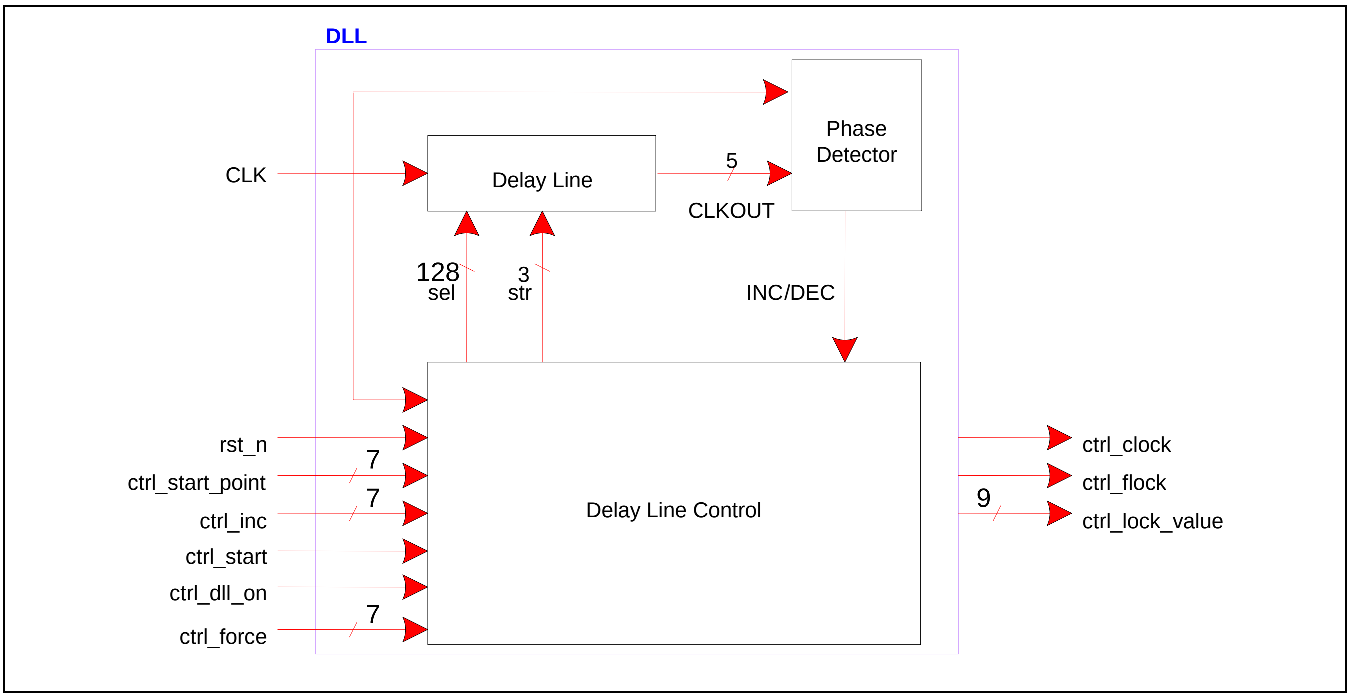

Figure 5-1. DLL Interface

DLL interface signals are shown in Figure 5-1. Delay line consists of 128 delay cells and they are controlled by 128-bit sel signals from delay line control logic. 3-bit str signals are used to control delay line in fine step resolution (fine step delay, tFS) with a quarter of the unit delay of the delay cell.

When ctrl_start becomes HIGH, initial start point and sampling point are loaded with the value of ctrl_start_point(the number of delay cells) and DLL starts running at this initial start point. Initial delay time is calculated by multiplying the unit delay of delay cell and this initial start point. Phase detector starts detecting lead or lag of CLKOUT(delayed multiple phase clock output) in comparison with CLK by sampling CLKOUT at rising edge of CLK. Sampling point is increased or decreased according to this lead or lag information from phase detector.

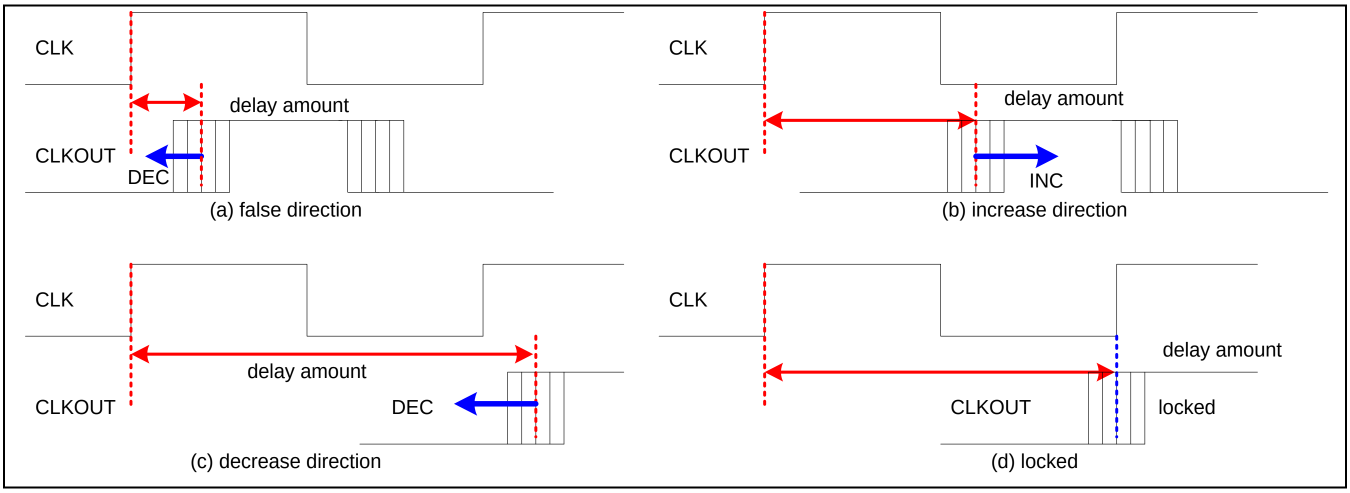

Figure 5-2. DLL lead/lag/lock Cases

Figure 5-2. illustrates lead/lag/lock cases of DLL. In the case of (a), sampling point is decreased to almost zero delay, and which is false direction. In this case, initial start point is added with ctrl_inc and this new value becomes new sampling point.

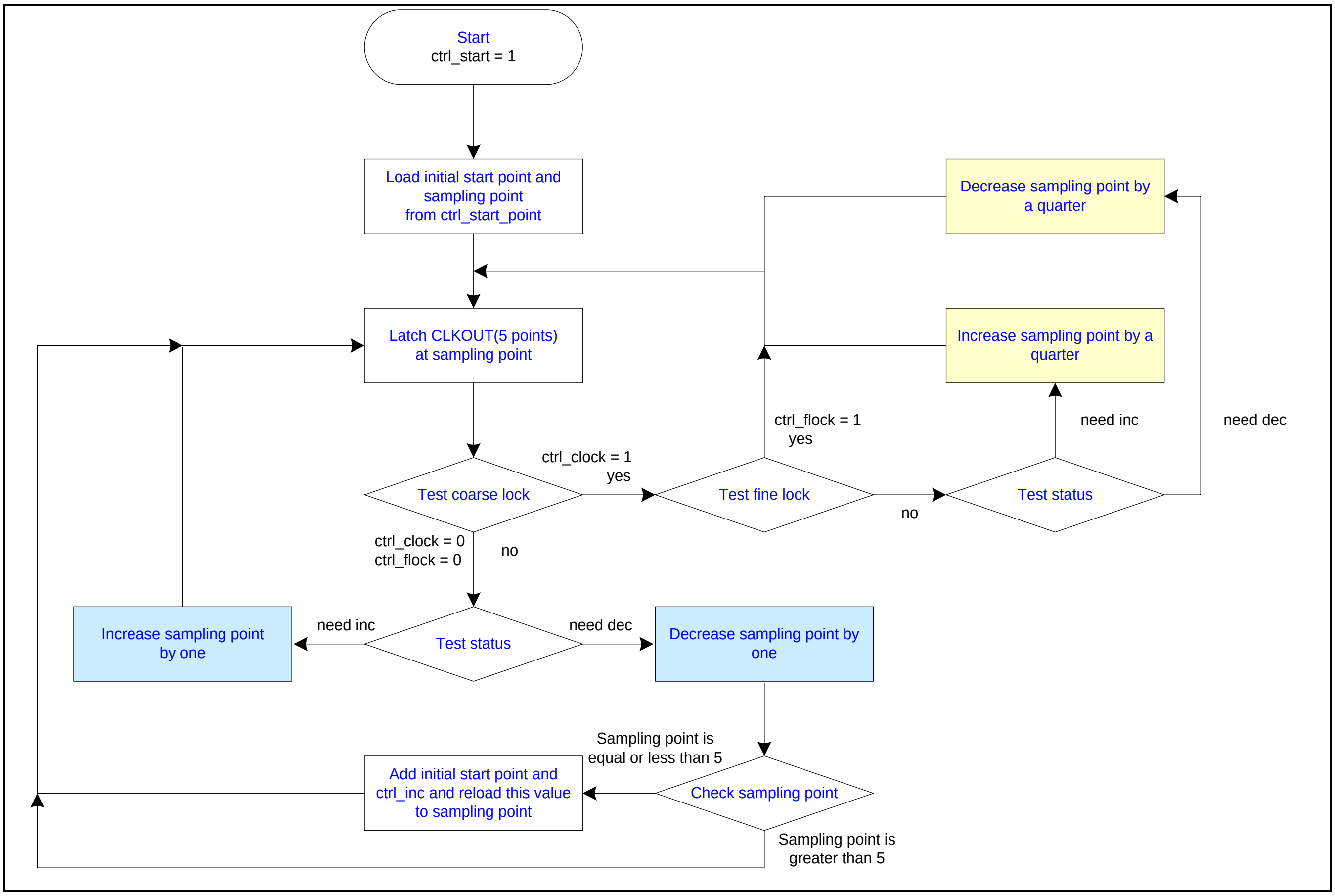
Start
ctrl_start = 1
Load initial start point and sampling point from ctrl_start_point
Latch CLKOUT(5 points) at sampling point
Test coarse lock
ctrl_clock = 1
yes
Test fine lock
ctrl_flock = 1
yes
no
Test status
need inc
need dec
Decrease sampling point by a quarter
Increase sampling point by a quarter
ctrl_clock = 0
ctrl_flock = 0
no
Test status
need inc
need dec
Increase sampling point by one
Decrease sampling point by one
Check sampling point
Sampling point is equal or less than 5
Sampling point is greater than 5
Add initial start point and ctrl_inc and reload this value to sampling point

Figure 5-3. DLL Lock Flow Chart

Figure 5-3. illustrates flow chart of DLL operation. Before ctrl_clock is HIGH, DLL controls delay line in the unit delay of delay cell. After ctrl_clock is High, DLL changes step resolution and enters fine tracing mode and delay line is controlled in a quarter of the unit delay of delay cell.

{ctrl_clock, ctrl_flock = 2'b00} : DLL is not locked.

{ctrl_clock, ctrl_flock = 2'b01} : Impossible value.

{ctrl_clock, ctrl_flock = 2'b10} : Locked in coarse lock stage.

{ctrl_clock, ctrl_flock = 2'b11} : Locked in fine lock stage.

During initialization, dfi_init_complete should be checked to confirm DLL lock before an update request which is used to propagate lock information to each data slice. ctrl_clock may repeat set and clear because of PLL jitter. Thus, it's not required to check ctrl_clock during memory access. During the normal operation, dfi_ctrlupd_req should be set and cleared after auto-refresh is started.

5.2 FREQUENCY RATIO CLOCK DEFINITION

The DFI clock(Controller clock) and PHY clock(clk2x) must be phase-aligned. The additional clock enable signal(=clk_en) will be internally used so that the rising edge of clk2x is aligned with the rising edge of DFI clock. "clk_en" should be always high at both rising edges of "DFI clock" and "clk2x". "clkm" has no relation with clk2x but will be same frequency(400~800MHz). If "clk2x" is under 400MHz, please refer to 8.2 LOW FREQUENCY OPERATION.

The PHY should maintain the information to preserve the timing relationships between commands and data. Therefore, for frequency ratio systems, the control signal interface, the write data interface and the read data enable signal will all be suffixed with a "_pN" where N is the phase number. As an example, for a 1:2 frequency ratio system, instead of a single dfi_address signal, there will be 2 signals: dfi_address_p0 and dfi_address_p1. The read data signal, read data valid and read data not valid signals will be suffixed with a "_wN" where N is the DFI data word.

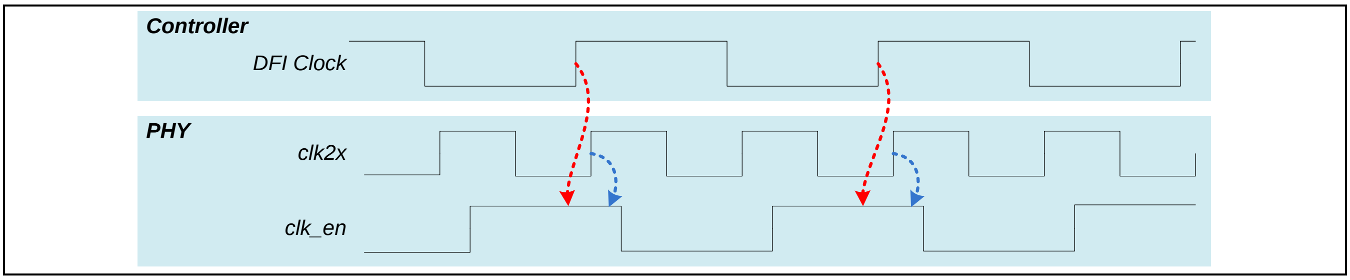

Figure 5-4. Timing Diagram between Controller Clock and PHY Clock(1:2 frequency ratio)

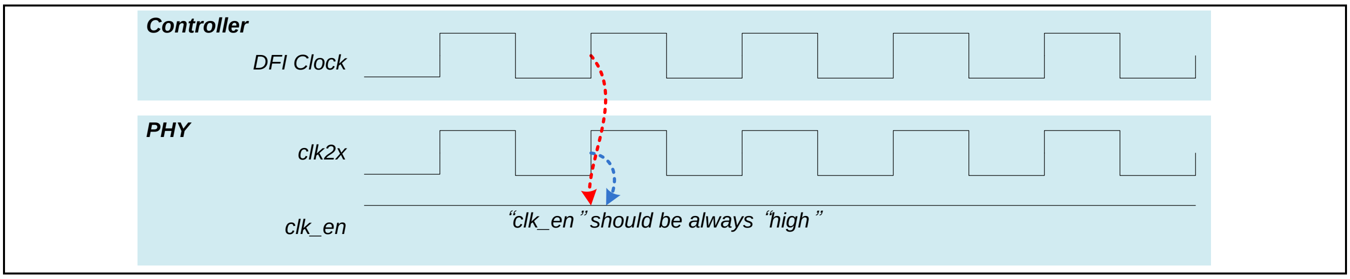

Figure 5-5. Timing Diagram between Controller Clock and PHY Clock(1:1 frequency ratio)

5.3 CONTROL PATH

Signal interface timing for the 1:2 frequency ratio control path is shown in Figure 5-6. and how the PHY in this system would interpret the DFI signals. In this example, a command is only sent on phase 0 and ODT information is provided on both phases.

Preliminary product information describes products that are in development, for which full characterization data and associated errata are not yet available. Specifications and information herein are subject to change without notice.

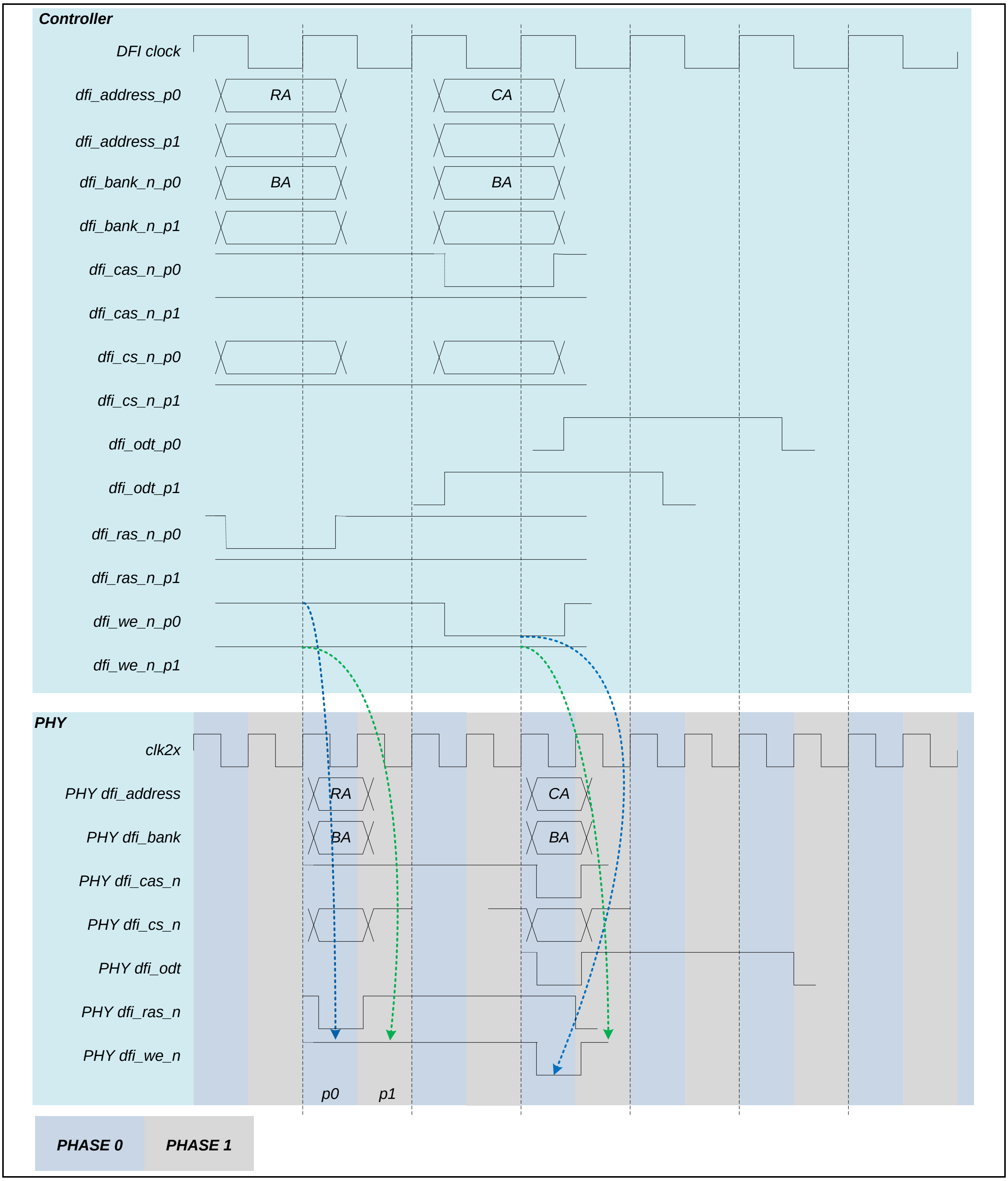

Figure 5-6. Timing Diagram for Frequency Ratio Control Path

5.4 DATA PATH

The timing parameters tphy_wrlat and tphy_wrdata define the delay from the write command to the dfi_wrdata_en_pN signal, and from the dfi_wrdata_en_pN signal to when data will be driven on the dfi_wrdata_pN signal respectively. These timing parameters are defined in terms of DFI PHY clocks.

- tphy_wrlat=WL-1, tphy_wrdata=2, for example if WL=4, tphy_wrlat=3, tphy_wrdata=2(LPDDR3)
- tphy_wrlat=WL-2, tphy_wrdata=2, for example if WL=8, tphy_wrlat=6, tphy_wrdata=2(DDR3)

Signal interface timing for data write is shown in *Figure 5-7*. dfi_wrdata_en_p0/p1 is enable signal to indicate dfi_wrdata_p0/p1 and dfi_mask_p0/p1 are valid.

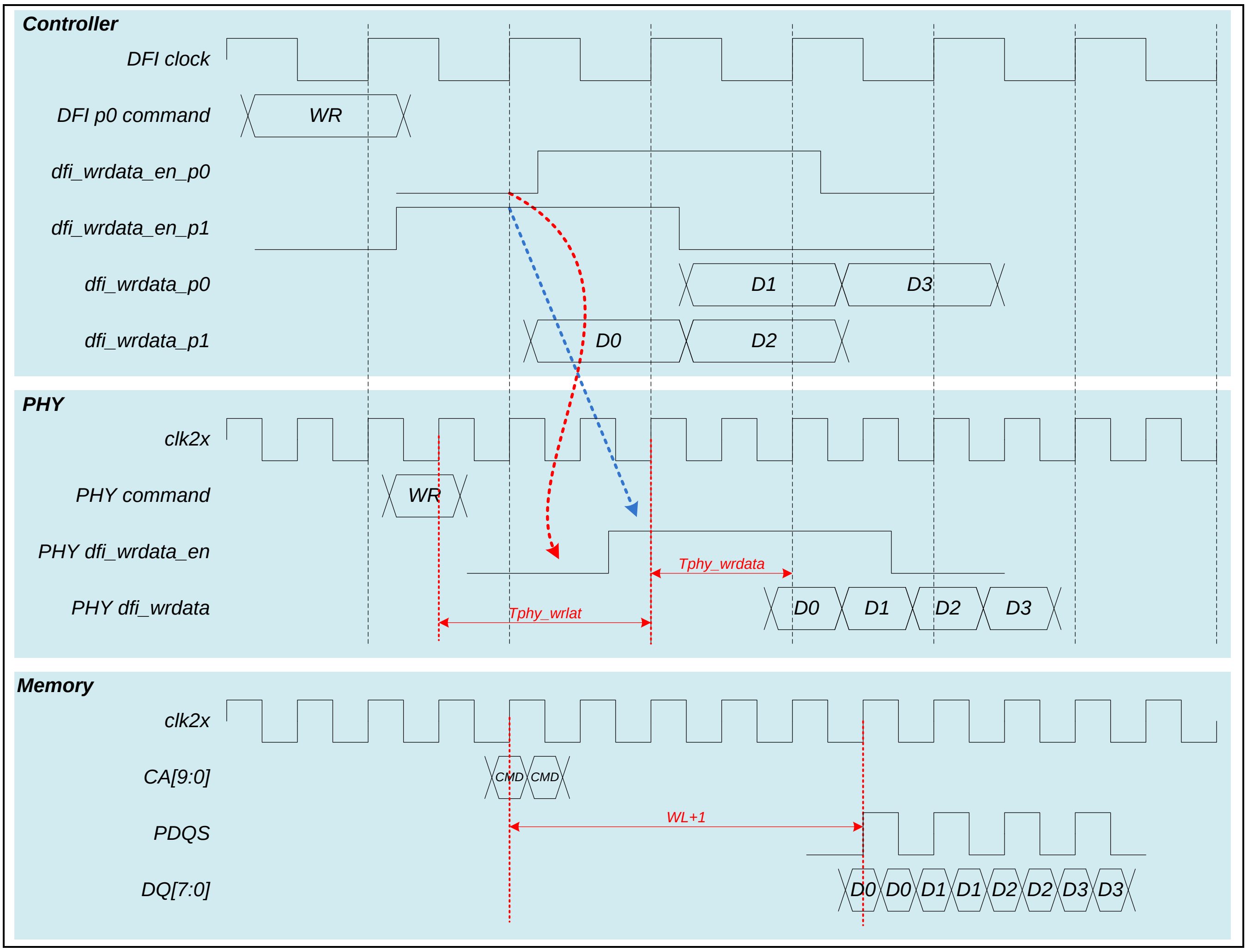

Figure 5-7. Data write Timing Diagram(1:2 frequency ratio, LPDDR3)

Preliminary product information describes products that are in development, for which full characterization data and associated errata are not yet available. Specifications and information herein are subject to change without notice.

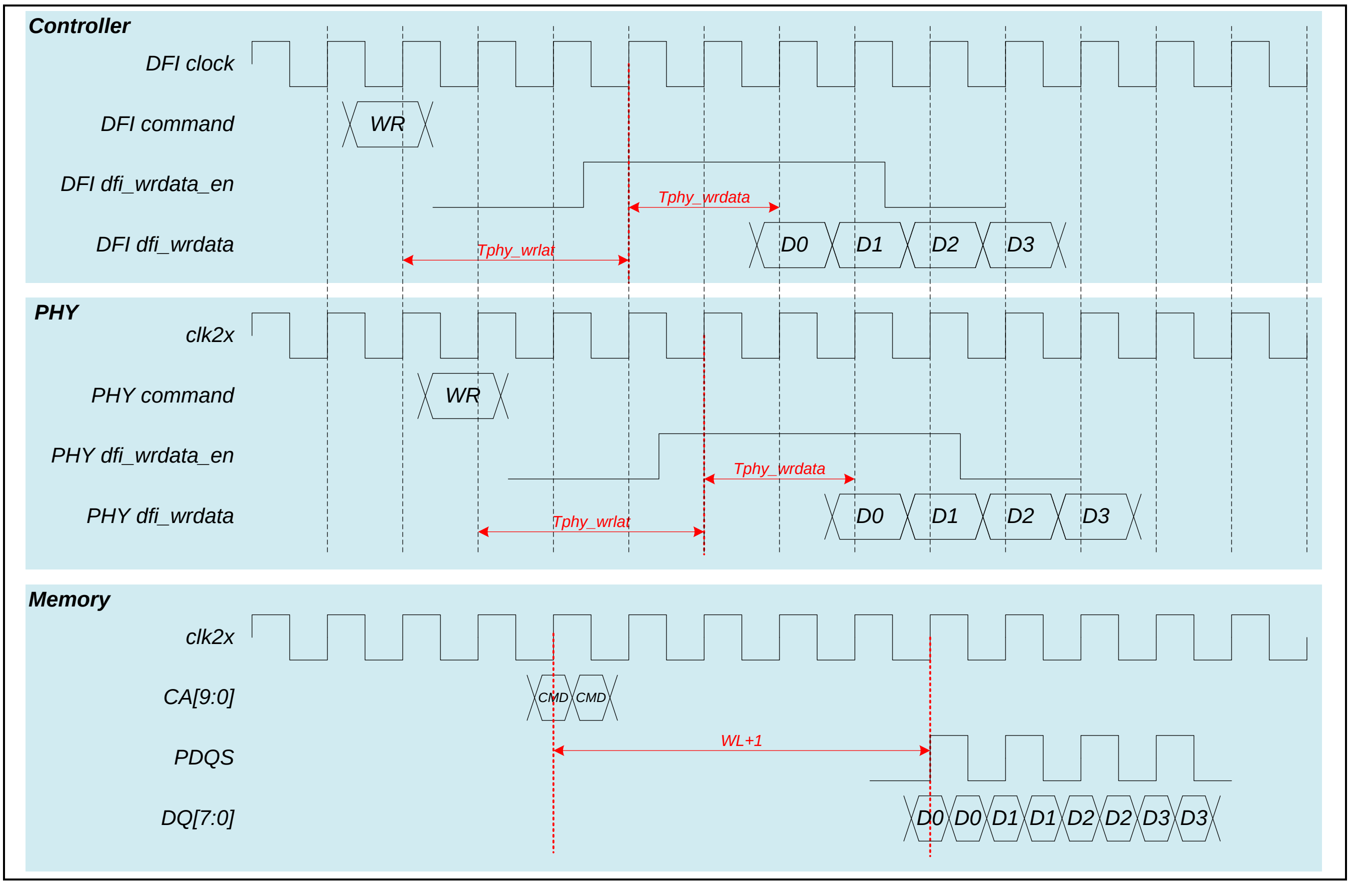

Figure 5-8. Data write Timing Diagram(1:1 frequency ratio, LPDDR3)

The timing parameters trddata_en and tphy_rdlat define the delay from the read command to the dfi_rddata_en_pN signal, and from the dfi_rddata_en_pN signal to when data will be returned on the dfi_rddata_wN bus, respectively. These timing parameters are defined in terms of DFI PHY clocks and are measured relative to how the PHY interprets the data.

- trddata_en=The value of "T_rddata_en" in PHY_CON18, tphy_rdlat=2, for example if T_rddata_en=9, trddata_en=9, thy_rdlat=2

Signal interface timing for data read is shown in *Figure 5-9*. After read command is issued, DQ and PDQS/NDQS are driven by the memory and DQ is stored in FIFO in read data path. After read data is stored in FIFO, when dfi_rddate_en_p0/p1 is issued from controller, the valid read data is driven during HIGH dfi_rddata_valid_w0/w1 signal at the rising edge of the clk signal.

NOTE: Please refer to "6. DQS Clean" for ctrl_gate_p0/p1 interface timing.

Caution: dfi_rddata_en_p0 and dfi_rddata_en_p1 should be enabled at the same time.

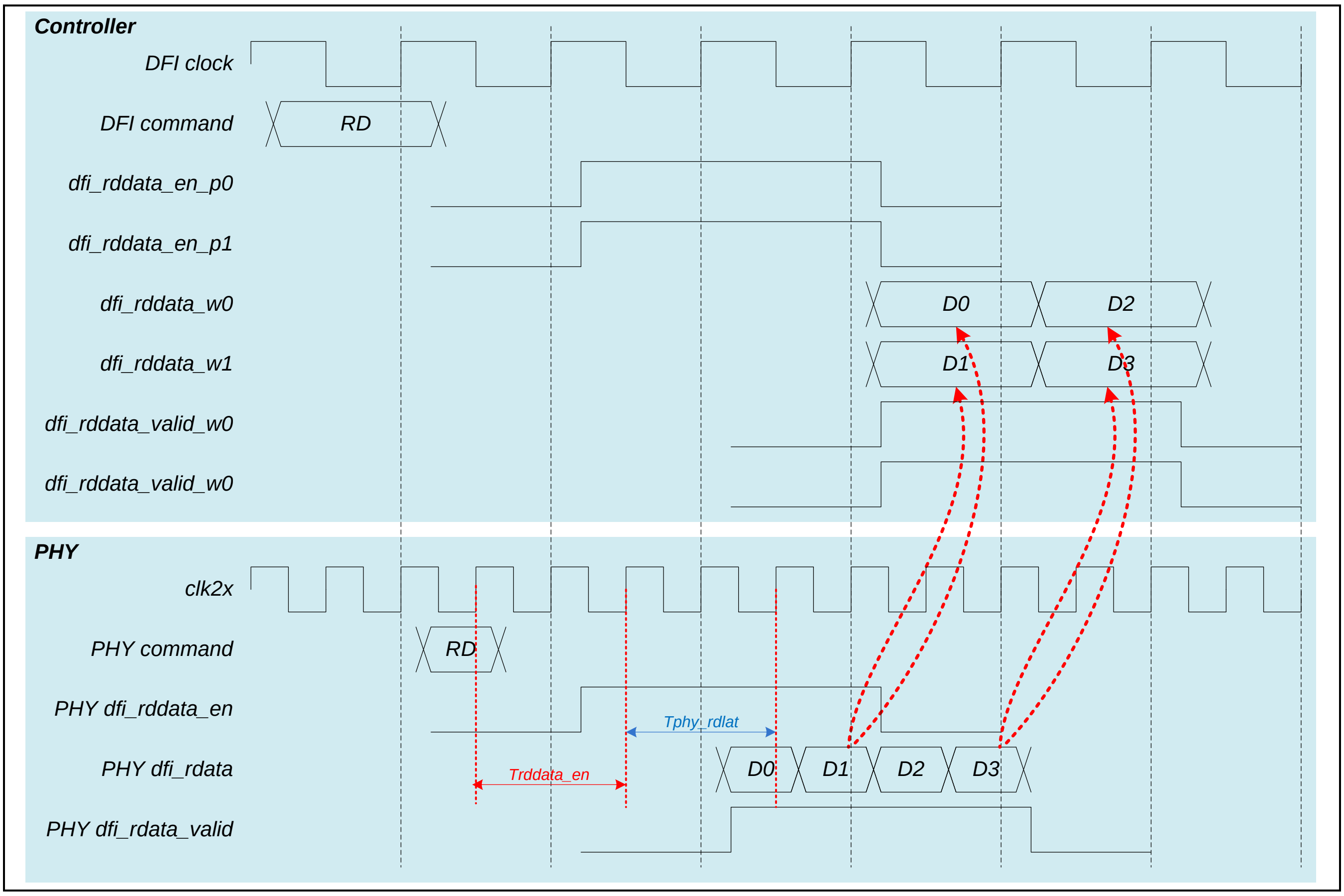

Figure 5-9. Data Read Timing Diagram(1:2 frequency ratio)

Figure 5-10. illustrates control signal and data read path. At the read path, DQS clean block is used to clean DQS signal and Delay line is used for 90° phase shift of DQS. If read command is issued at A point, read command is driven and memory drives DQ and DQS after CL(CAS Latency) at B point. DQ and DQS experience board and IO input delay and arrive at C point. Delay line execute 90° phase shift of DQS and shifted DQS is used as a clock signal to capture DQ in FIFO. After that dfi_rddata_en_p0/p1 is driven HIGH by the controller, the FIFO outputs valid read data at E point.

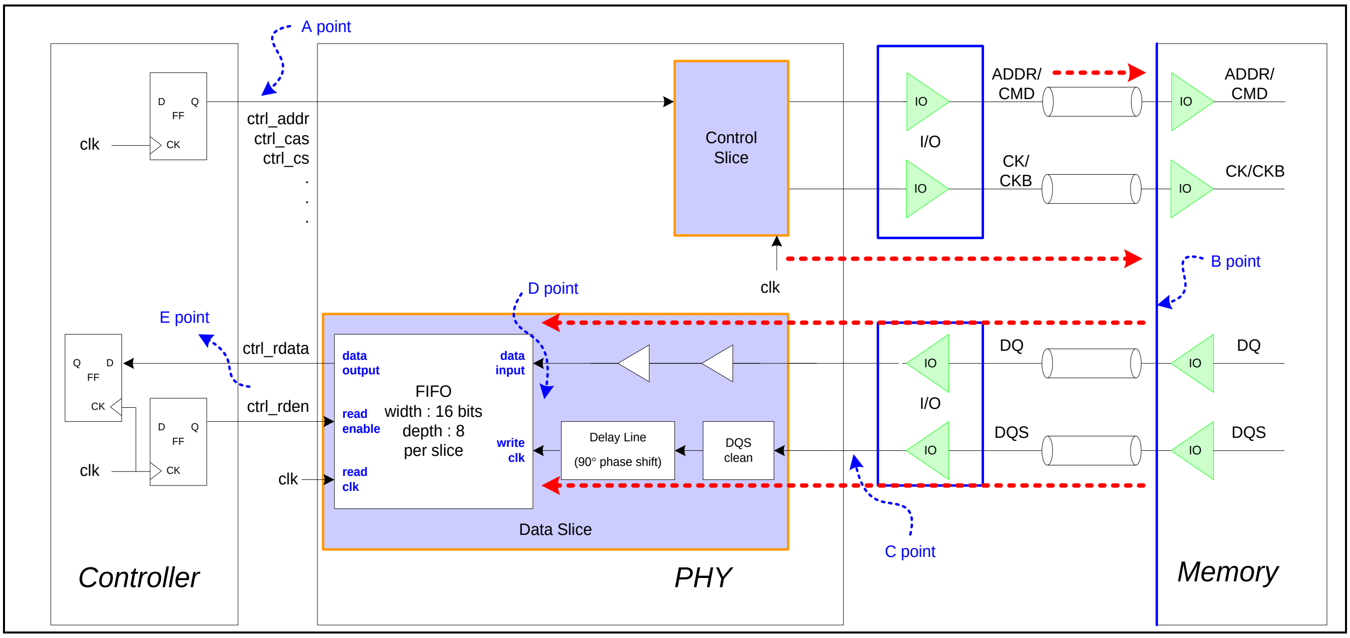

Figure 5-10. Read Data Path

Figure 5-11. illustrates the full timing diagram of command and data path for memory read. The deterministic timing values for the read path are latency to generate command (1 clock cycle), CL(CAS Latency) and data write time in FIFO(1 clock cycle). But the propagation delay of command (from PHY to memory) and DQ/DQS (from memory to PHY) varies according to the board designs and operating condition. And also internal logic delay in Figure 5-11. is also different according to operating condition. For this reason, read data arrival time is quite different according to designs and environment and deterministic read timing calculation is difficult. To overcome this issue, FIFO of 8-depth is used in read path and data read timing can be programmed.

To consider the read enable timing(when to issue dfi_rddata_en_p0/p1 signal, "Trddata_en") for minimum read latency, PHY provides the optimal read timing after the read leveling is finished(Refer to PHY_CON18). To consider Voltage/ Temperature variations during operation, the maximum value of "Trddata_en" which has got from PHY_CON18 should be added by 1. For example, if T0_rddata_en=10, T1_rddata_en=11, T2_rddata_en=10 and T3_rddata_en=11, Trddata_en should be 12.

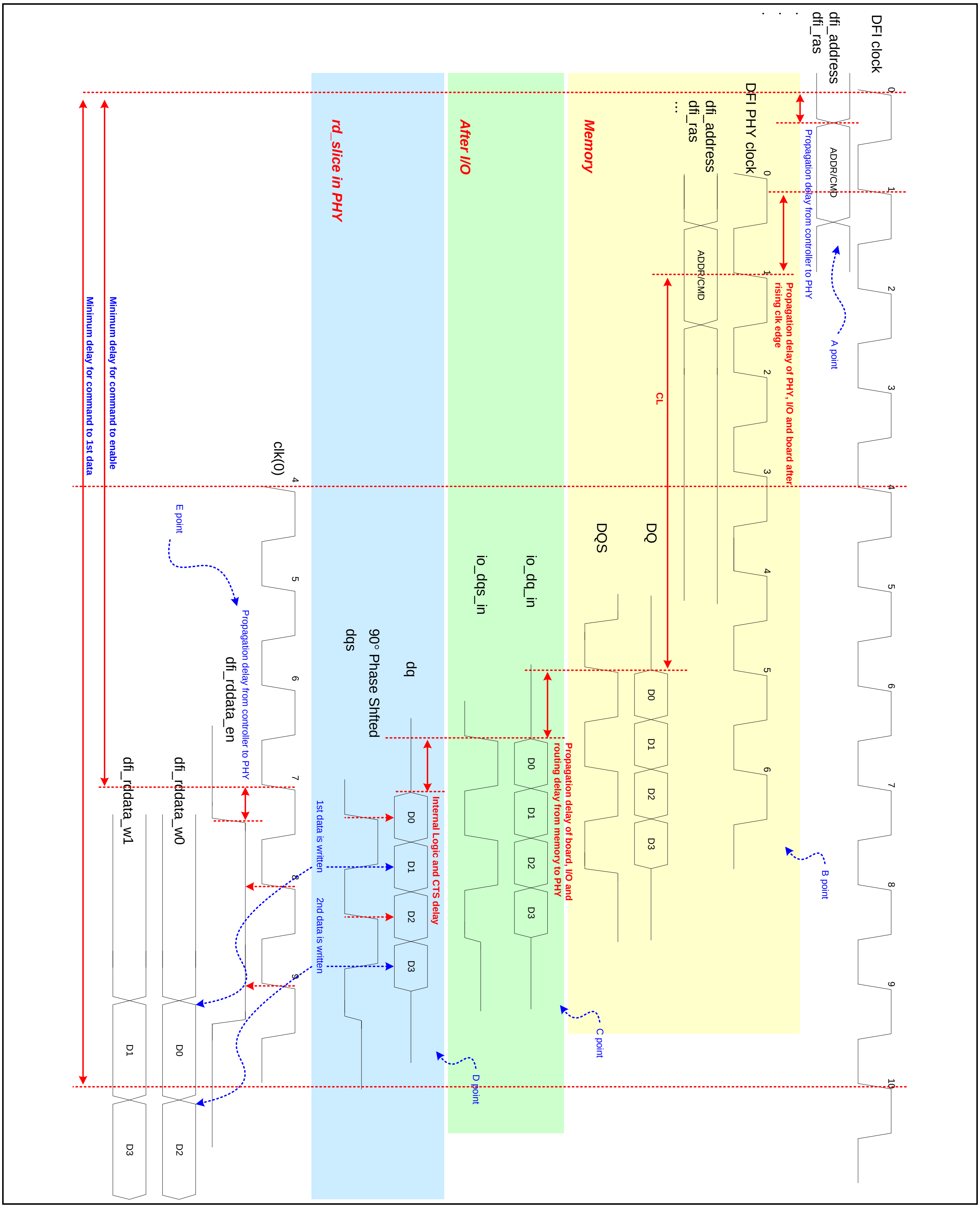

Figure 5-11. Full Read Path Timing Diagram

6 DQS Clean

DQS cleaning is needed to remove high-Z state of DQS during read, but there is a lot of variation in DQS. We recommended that the pull-down mode in DQS(PHY_CON14 [3:0]=4'hF) should be used to remove high-Z state of DQS.

When it is in the pull-down mode, the duration of HIGH in "ctrl_gate_p0 /p1" should be extended to compensate the variation of DQS with ctrl_shgate = 0(Refer to Table 6-1). But the duration of HIGH in "ctrl_gate_p0/p1" should be always [(the burst length of read transaction)/2+1] in case of DDR3. tRTW(=Read to Write turn around time) should be carefully considered as the duration of "ctrl_gate_p0/p1" is extended. Controller should have a programmable option to extend tRTW. "Delay line" in Figure 6-1 is controlled by ctrl_offsetc and ctrl_shiftc . Controller should issue "ctrl_gate_p0/p1" in the following way.

- "ctrl_gate_p0/p1" should be issued after RL+1 from read command issue from controller. (LPDDR3)
- "ctrl_gate_p0/p1" should be issued after RL-1 from read command issue from controller. (DDR3)

Warning: The minimum time from the burst read command to the burst write command is defined by tRTW(=read-to-write-turnaround-time), which is 4 clocks in case of BL = 4 operation, 6 clocks in case of BL = 8 operation. Controller should have a programmable option to extend "tRTW" up to 14.

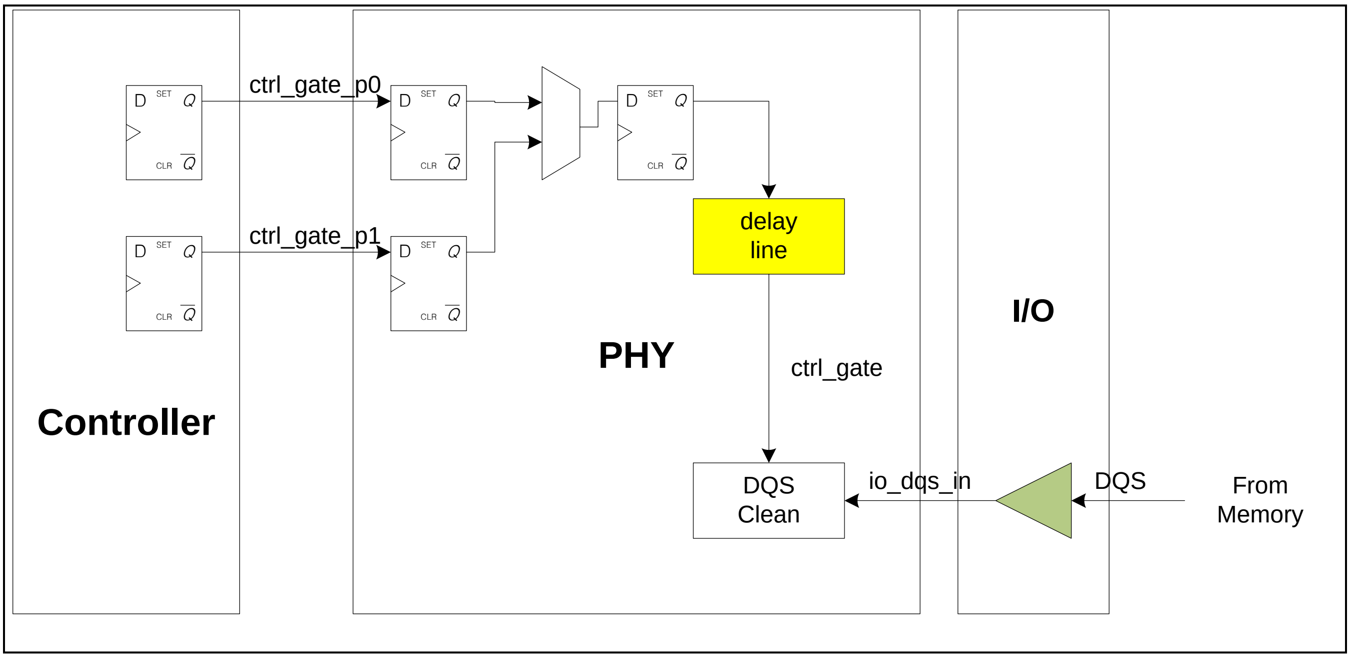

Figure 6-1. Block Diagram of GATE Signal Path

Preliminary product information describes products that are in development, for which full characterization data and associated errata are not yet available. Specifications and information herein are subject to change without notice.

If Gate Leveling is applied, "ctrl_gate_p0/p1" don't need to be issued from controller, but the setting should be ctrl_shgate =1 and ctrl_atgate=1.

"ctrl_gate_p0/p1" can be generated internally by setting "ctrl_atgate=1" in PHY_CON0 register. When ctrl_atgate is high, PHY can clean DQS without any intervention from controller. But there are some limitations to use this mode. The setting of burst length(=BL) should be also changed for DQS cleaning if there is any change of setting in memory. So it is recommended that "ctrl_gate_p0/p1" should be generated from controller to provide the dynamic controllability.

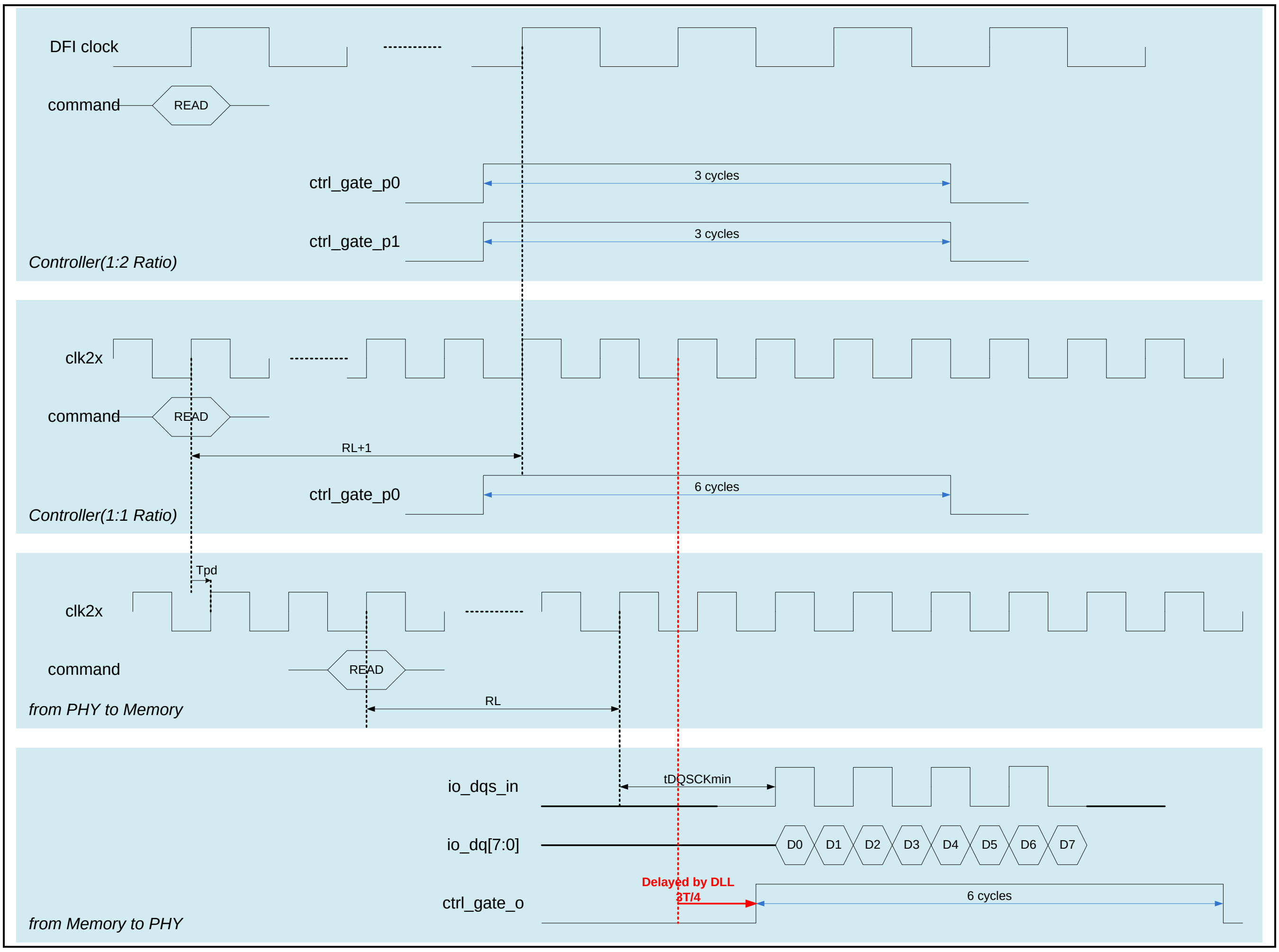

Figure 6-2. GATE Signal when tDQSCK=2.5ns(LPDDR3 800MHz)

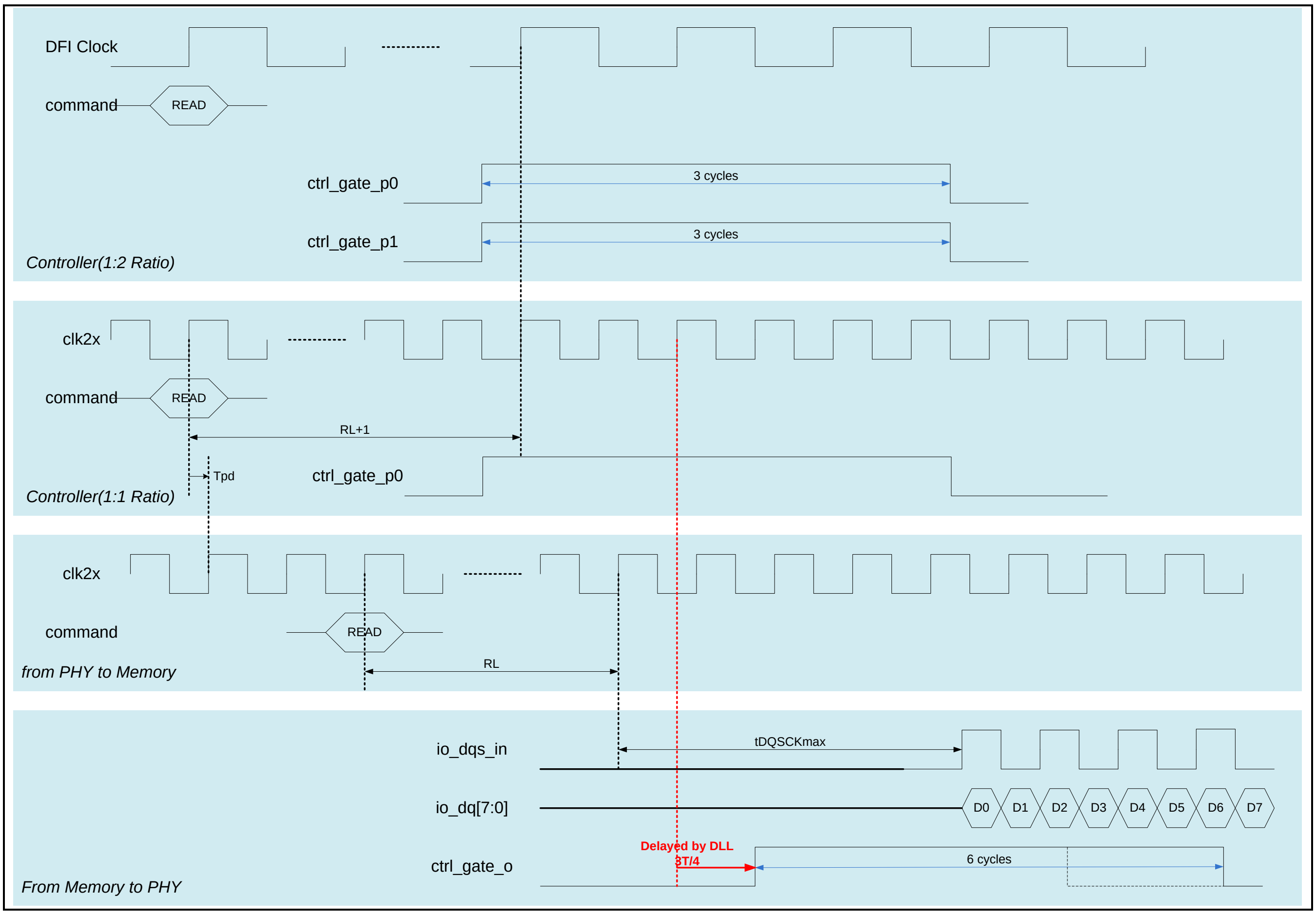

Figure 6-3. GATE Signal when tDQSCK=5.5ns(LPDDR3 800MHz)

Tpd is the delay in I/O, Package, Board(*Figure 6-2, Figure 6-3*). This delay will be very variant depending on PVT conditions. Those variations should be compensated by the enough duration of GATE.

Table 6-1. GATE Signal Guide according to Operation frequency(LPDDR3).

Memory Type	Frequency (MHz)	GATE Assertion cycle form READ Command	The duration of GATE
LPDDR3	800	RL+1+3T/4	BL/2+2+Tpd
LPDDR3	667	RL+1+T/2	BL/2+2+Tpd
LPDDR2/3	533	RL+1+T/4	BL/2+1+Tpd
LPDDR2/3	400	RL+3T/4	BL/2+1+Tpd

NOTE: Suppose that Tpd is about 0.3ns in *Figure 6-2, Figure 6-3* and *Table 6-1* for easy explanations.

NOTE: Usually Tpd will be increased in real environment. If it is increased, please increase the duration of GATE depending on the value of Tpd.

Caution: If Gate Leveling is used, The duration of GATE should be "BL/2 - 1" (DDR3, 800MHz)

7 Test Mode

11 operation modes are supported and controlled by mode pins. The other operation modes except normal function mode are used to test the PHY. Figure 7-1. shows that test signals should be allocated to external pins–multiplexed with functional pins for test mode.

Table 7-1. Operation Mode

mode_phy	mode_nand	mode_scan	mode_mux	mode_run	mode_highz	ctrl_fnc_fb	Mode
0	0	0	0	000	0	000	Normal
0	0	0	0	000	0	010	ATE EFNC
0	0	0	0	000	0	011	ATE IFNC *
0	0	0	1	000	0	Don't care	MUX
0	0	1	0	000	0	Don't care	SCAN
0	1	0	0	000	0	Don't care	Nand-Tree
1	0	0	0	000	0	Don't care	I/O
0	0	0	0	000	1	Don't care	High-Z
1	0	0	0	001	0	Don't care	ATE ERF
1	0	0	0	010	0	Don't care	ATE IRF *
1	0	0	0	100	0	Don't care	ATE IWF *
1	0	0	0	101	0	Don't care	ZQ I/O

If mode_highz is set, all I/O outputs are disabled. mode_highz can be set when PHY is in ATE IFNC, BRD IRF, BRD IWF, ATE IRF or ATE IWF mode.

For example, if mode_highz is set when PHY is in ATE IRF mode, ATE IRF can be executed because only the I/O outputs are disabled and ATE IRF is a test without I/O.

Table 7-2. Mode Description

Num	Mode	Description	Purpose
1	Normal	Normal operation	Normal operation
2	ATE EFNC	ATE External Function Feedback Test	For ATE test vector with CPU,I/O and MEMCON
3	ATE IFNC	ATE Internal Function Feedback Test	For ATE test vector with CPU and MEMCON
4	MUX	MUX Mode	To use I/O not for LPDDR but for other

			purpose
5	SCAN	ATE SCAN Test Mode	For ATE SCAN test vector
6	Nand-Tree	ATE Nand-Tree Test Mode	For ATE Nand-Tree test vector(VIL/VIH)
7	I/O	ATE I/O Test Mode	For ATE I/O Test vector(VOL/VOH)
8	ATE ERF	ATE External Read Feedback Test	For ATE PHY read test with I/O
9	ATE IRF	ATE Internal Read Feedback Test	For ATE PHY read test without I/O
10	High-Z	ATE High-Z Test Mode	For ATE SIP/MCP memory Test
11	ATE IWF	ATE Internal Write Feedback Test	For ATE PHY write test without I/O

NOTE: mode_highz can be set in internal Feedback Tests(3,5,6,13 and 15 operation mode)

NOTE: During the Nand-Tree mode, "rst_n" = 0(the clock running of "clk2x" is needed to propagate rst_n=0), ~~"mode_high_z" = 0, "ctrl_pulld_dq[NS-1:0]" = 'h0, "ctrl_pulld_dqs[NS-1:0]" = ’h0,~~ "test_ext_ cmosrcv" = 1'b0 or 1'b1. "test_ext_en" = 1'b0, "ctrl_pd[NS:0]" = 'h0. (NS means the number of slice)

NOTE: During I/O Test mode, "rst_n" = 0(the clock running of "clk2x" is needed to propagate rst_n=0), ~~"mode_high_z" = 0, "ctrl_pulld_dq[NS-1:0]" = 'h0, "ctrl_pulld_dqs[NS-1:0]" = 'h0,~~ "test_ext_ cmosrcv" = 1'b0 or 1'b1. (NS means the number of slice)

Table 7-3. Required Vector List

Vector	Speed	Requirement
External Read Feedback Test	Low-speed	Recommended*
Internal Read Feedback Test	At-speed	Mandatory
Internal Write Feedback Test	At-speed	Mandatory
External Function Feedback Test	Low-speed	Recommended*
Internal Function Feedback Test	At-speed	Recommended
SCAN Test Mode	Low-speed	Mandatory
Nand-Tree Test Mode	Low-speed	Mandatory
I/O Test, ZQ I/O Test	Low-speed	Mandatory
High-Z Test Mode	Low-speed	Mandatory

NOTE: Speed is limited by the load of test board and equipment.

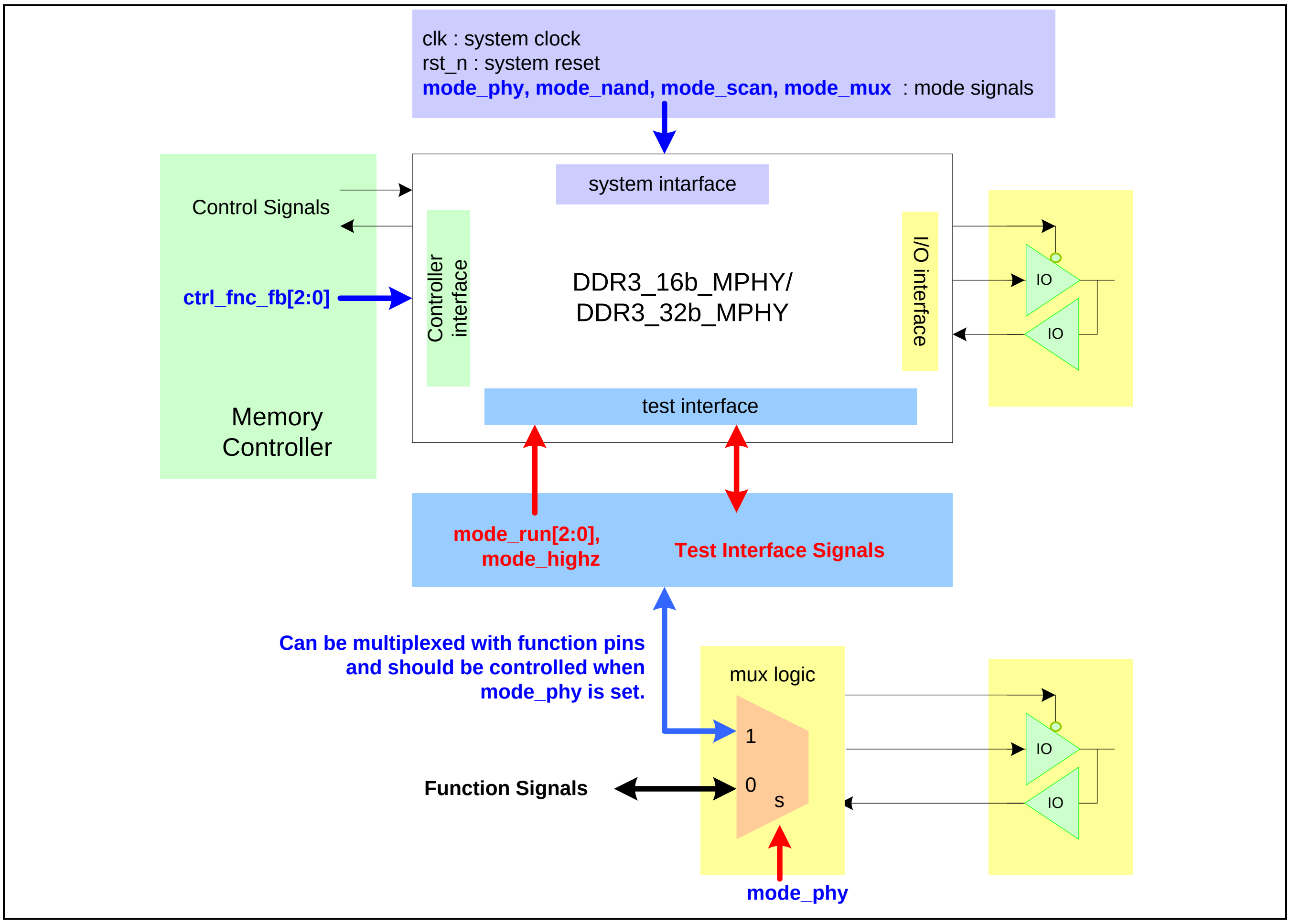

Figure 7-1. Test Pin Allocation

For test operation modes, some signals should be controlled by external pin and shown in Table 7-2. "M" means 'mandatory', i.e. should be assigned to external pins. "R" means 'Recommend', i.e. recommended to be assigned to external pins.

Table 7-4. Test Pin Description

	Port	I/O	Description	Severity
Mode Signal	test_ext_dfdqs	I	tie to 1 for differential.	R
	test_ext_cmosrcv	I	1'b0 : Differential receiver mode for high speed operation 1'b1 : CMOS receiver mode for low speed operation (< 200MHz)	R
	test_ext_lpddr2	I	tie to 1 for LPDDR2/LPDDR3 tie to 0 for DDR2/DDR3	R
	mode_run[2:0]	I	test mode control	M
ZQ Control	test_ext_zq_end	O	ZQ Calibration Done	M

	Port	I/O	Description	Severity
Signal				
Feedback Control Signal	test_ext_lock_value[8:0]	O	Lock Value	M
	test_ext_start_point[6:0]	I	If not assigned, tie to 8'h14.	R
	test_ext_inc[6:0]	I	If not assigned, tie to 8'h14.	R
	test_ext_clock	O	Coarse lock signal	R
	test_ext_flock	O	Fine lock signal	R
	test_ext_locked	O	Lock signal	M
	test_ext_init_complete	O	Same as "dfi_init_complete"	M
	test_ext_ref[3:0]	I	If not assigned, tie to 4'h1. test_ext_ref[0] controls "byte_level_en"(PHY_CON[13]) during feedback test.	R
	test_ext_shiftc[4:0]	I	Needed for margin test. (Default:5'h10)	R
	test_ext_offsetc[7:0]	I	Needed for margin test. (Default:0)	M
	test_ext_offsetd[7:0]	I	Needed for margin test. (Default:'h8)	R
	test_ext_offsetr[7:0]	I	Needed for margin test. (Default:'h8)	R
	test_ext_offsetw[7:0]	I	Needed for margin test. (Default:'h8)	R
	test_resync	I	For Debug,	R
	test_start[4:0]	I	3-bit for 16-bit for PHY	M
	test_err[4:0]	O	3-bit for 16-bit for PHY	M
	test_oky[4:0]	O	3-bit for 16-bit for PHY	M
	test_ext_mode[3:0]	I	Test Mode Select Signal	M
	test_ext_rdlvl_en	I	Read Leveling Enable	M
	test_ext_rdlvl_wr_en	I	Write Training Enable	M
	test_ext_gatelvl_en	I	Gate Training Enable	M
	test_ext_rdlvl_vwml[31:0]	O	Please don't assign.	.
	text_ext_rdlvl_vwmc[31:0]	O	Center value for Valid Window Margin	M
	test_ext_rdlvl_vwmr[31:0]	O	Please don't assign.	.
	test_ext_rdlvl_incr_adj[3:0]	I	If not assigned, tie to 'h4	M
IO Control Signal	test_ext_enable	I		M
	test_ext_output	I		M
	test_ext_read	I		M
Scan Signal	test_se	I		M
	test_si[19:0]	I	16-bit for 16-bit PHY	M
	test_so[19:0]	O	16-bit for 16-bit PHY	M

7.1 PHY FEEDBACK TEST

Feedback test mode is used for at-speed test. For this test, PLL should be turned on. After PLL is turned on and DLL is locked, feedback test is executed when test_start is set. If feedback test is done without any error, test_oky pin is set. Otherwise, test_err is set and test is stopped. PHY feedback test has two modes; internal and external test mode. Because data and strobe signals go out to I/O and come back through I/O, I/O can be tested at at-speed but the test speed is influenced by the load condition of the ATE. To remove the influence of the ATE's load condition, internal feedback test mode is supported and signals are routed internally. ZQ Calibration should be tested during external feedback test mode or I/O mode. If the external test is doing at the low frequency (<50MHz), “test_ext_dll_on” should be low. The DLL lock procedure can be removed to decrease a test time in the dll-off mode.

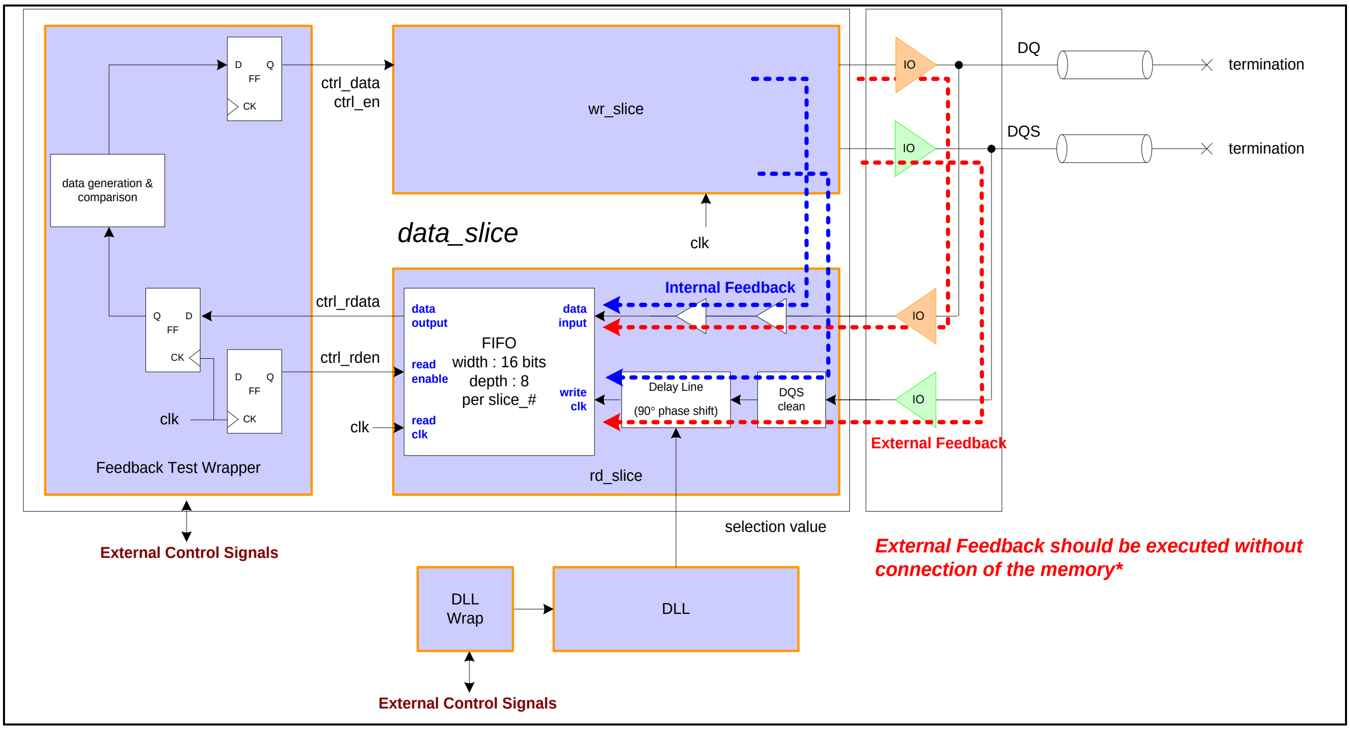

Figure 7-2. PHY Feedback Test Scheme

- Feedback Test Procedure
  - Set "test_ext_gatelvl_en"=1 for Gate Leveling, "test_ext_rdlvl_en"=1 for Read Feedback Test and "test_ext_rdlvl_wr_en"=1 for Write Feedback Test
    - If you don't use "Gate leveling", you can skip Gate leveling test by disabling "test_ext_gatelvl_en".
    - If you don't use "Read Calibration", you can skip Read Calibration test by disabling "test_ext_rdlvl_en"
    - If you don't use "Write Calibration", you can skip Write Calibration test by disabling "test_ext_rdlvl_wr_en"
  - Set "test_ext_mode", "mode_*" and so on (Please refer to Figure 7-3, Figure 7-4 for more details)
  - Reset(rst_n) is released.
    - The reset value of the internal registers in PHY should be used during Feedback, so be careful to keep the reset value as it is. For example, Feedback test may fail due to the unexpected behavior if there is any unknown input to PHY.

- Enable test_start until test_okay = 1.
- Check test_okay or test_err after about 8000 ~ 12000 cycles (refer to the following figures).

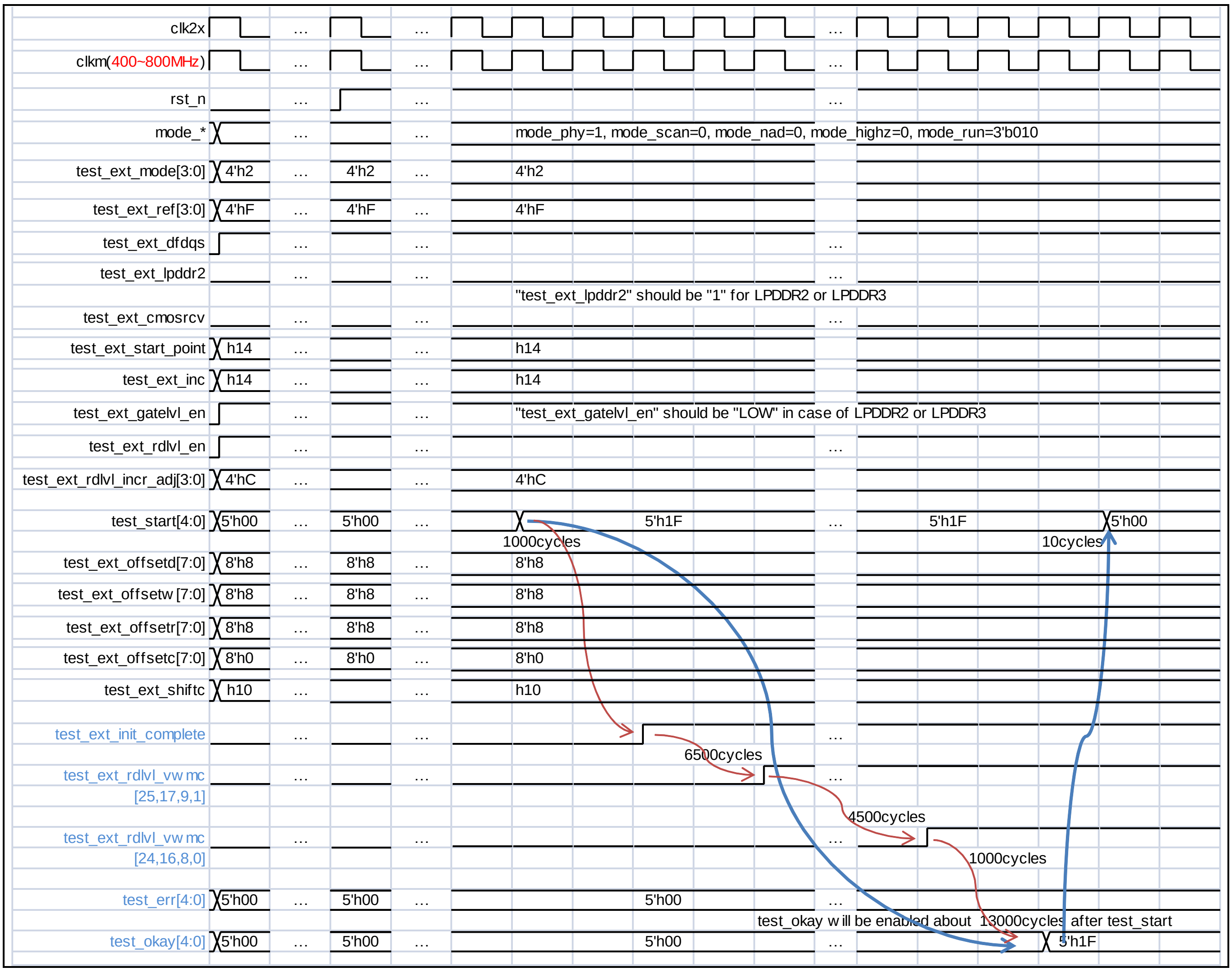

Figure 7-3. PHY Read Feedback

Caution: "dfi_freq_ratio" should be tied as "2'b01" to do feedback test.
Please skip Gate leveling if using LPDDR3 or LPDDR2 only.
If using DDR3, We recommend that Read Feedback Test should be separated as two vectors. Two vectors will be "Read Feedback" without Gate Leveling and "Read Feedback" with Gate Leveling.
Gate Leveling Read Feedback can be finished by checking "test_ext_rdlvl_vwmc[25,17,9,1]".

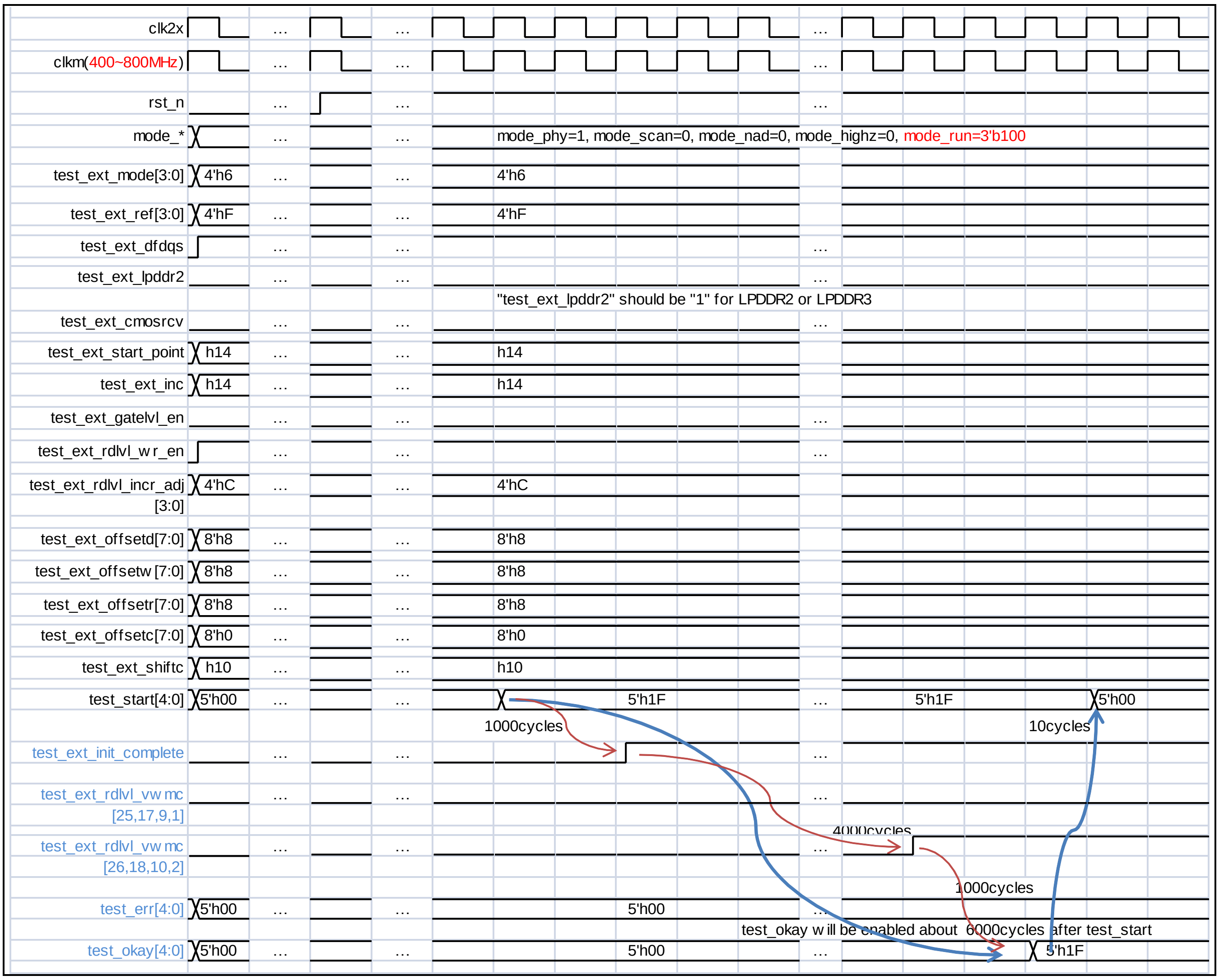

Figure 7-4. PHY Write Feedback

Caution: Any leveling functions can't be supported during External feedback
It should be "test_ext_gatelvl_en=0", "test_ext_rdlvl_en=0", "test_ext_rdlvl_wr_en=0" during External feedback

7.2 FNC FEEDBACK TEST

Function(FNC) feedback test mode is used for at-speed test with memory controller and CPU. The difference from PHY feedback test is tested with other blocks and control signals of PHY are controlled by memory controller. FNC feedback test mode has two modes; internal and external test mode. The important feature of this test mode is that "write data" generated by processor is automatically written to read FIFO. Therefore, if write data exceeds four-beat, the rest of the data will be over-written. For this reason, test vector should be programmed with taking this characteristic in mind. The four-beat burst access should be issued in FNC mode because DM(Data Mask) isn't supported.

Caution: Set DQS pull-down mode.
Set RL(Read Latency) by using the value of WL(Write Latency).
Leveling and Training functions is not supported.

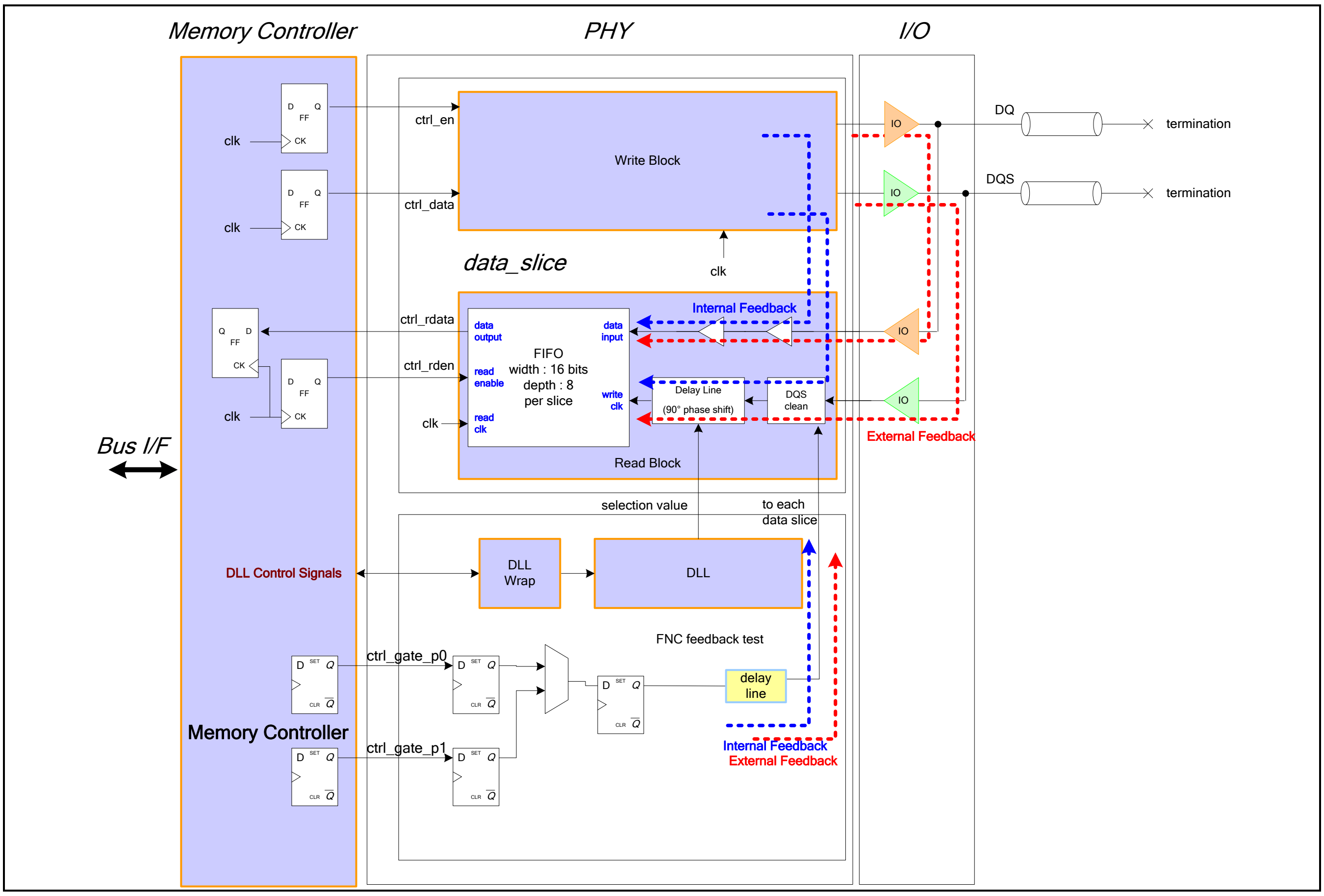

Figure 7-5. FNC Feedback Test Scheme

7.3 ZQ I/O TEST

ZQ I/O should be tested under 400MHz.

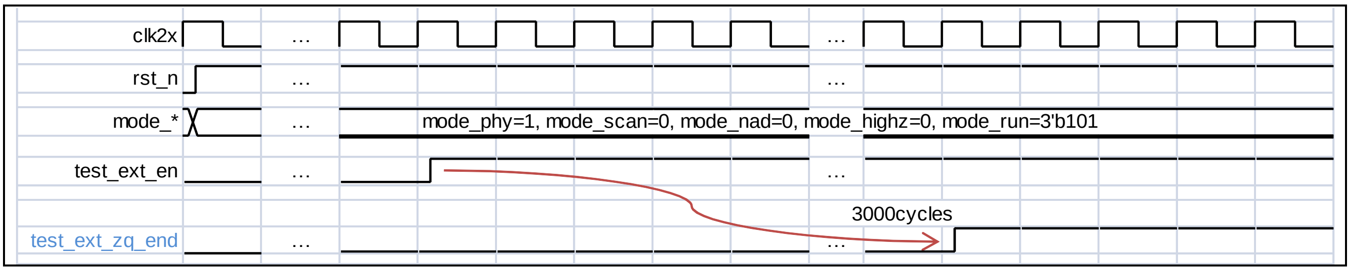
clk2x
rst_n
mode_*
mode_phy=1, mode_scan=0, mode_nad=0, mode_highz=0, mode_run=3'b101
test_ext_en
3000cycles
test_ext_zq_end
…

Figure 7-6. The example of ZQ I/O Test

7.4 I/O TEST

I/O test mode is used to test the output characteristics of I/Os. I/O test mode is controlled by G2 signals (refer to Table 7-2.). The I/Os that should be tested during I/O TEST will be ADCT[15:0], RAS, CAS, WEN, CKE, CK[1:0], GATEO, DM[3:0]

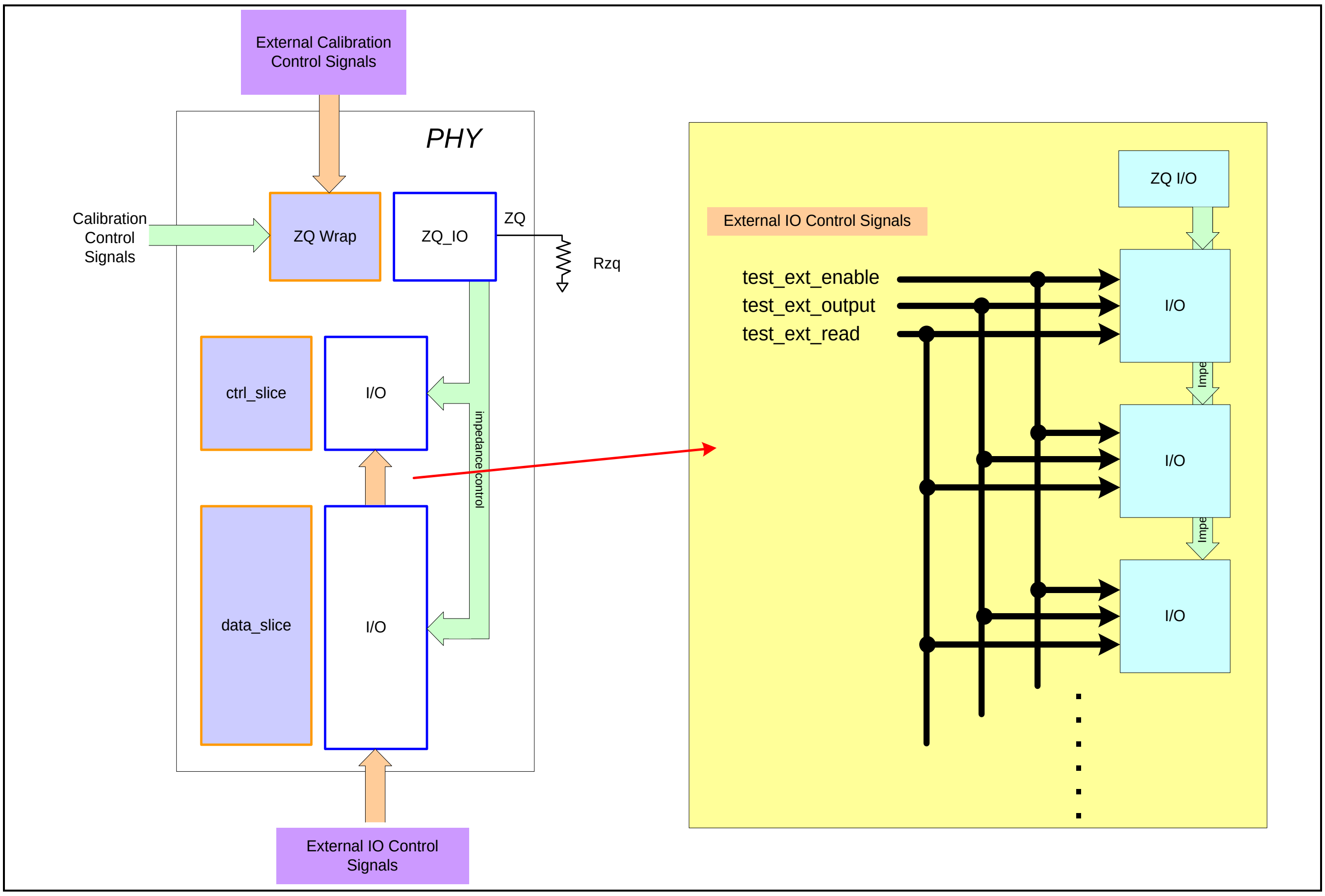

Figure 7-7. I/O Test

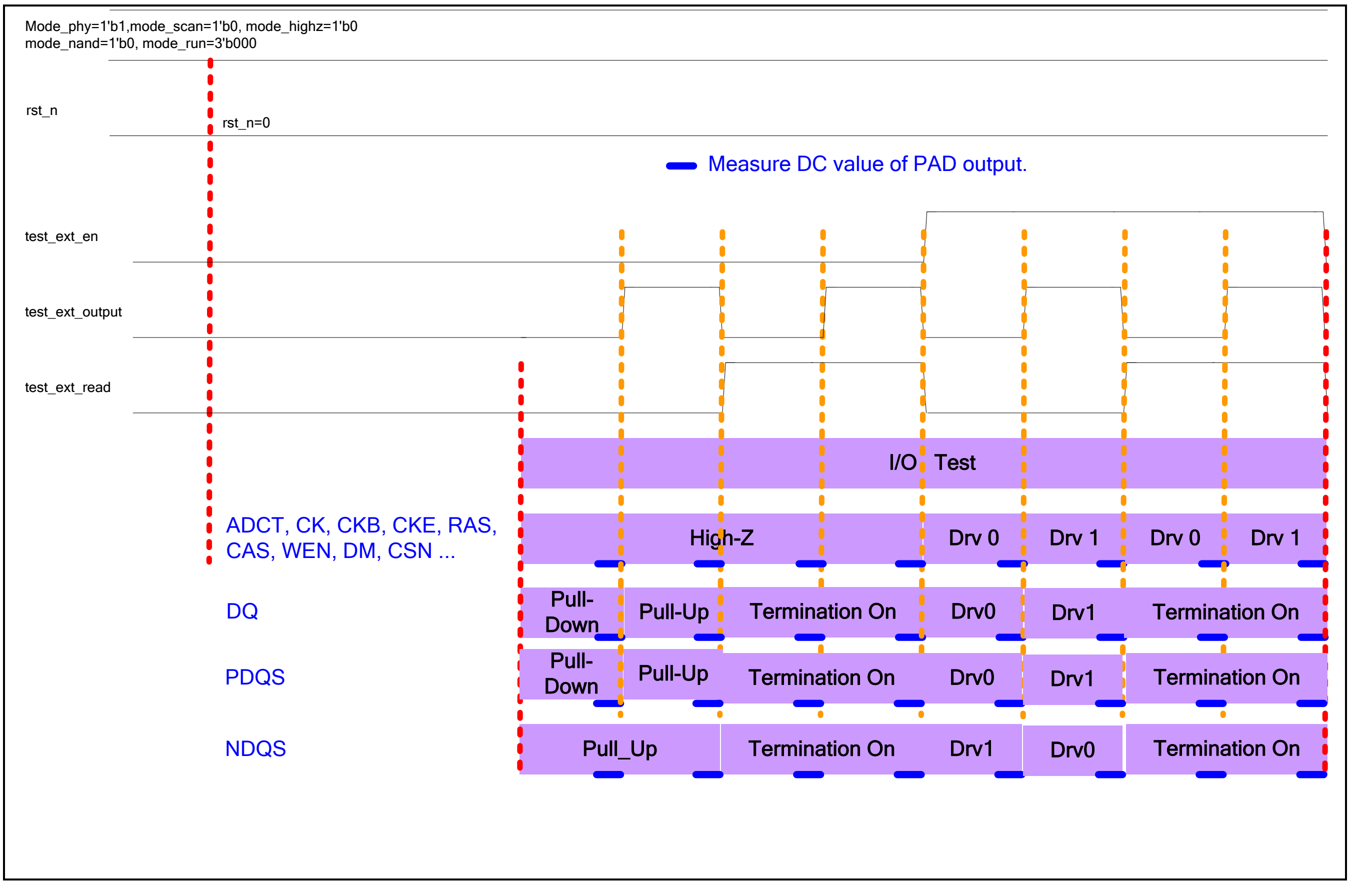

Figure 7-8. The Example of I/O Test

Caution: "rst_n" = 0(the clock running of "clk2x" is needed to propagate rst_n=0),

7.5 SCAN TEST

Scan test will cooperate with the top scan test and system clock is used for scan clock. Scan test should be done with full-chip scan test. (scan test mode of PHY should be set when full-chip is in scan mode) In Scan mode, PO(Primary Output) such as ADCT, BANK, CS, ODT, RAS, CAS, WE, CKE, DQ, DQS, DM should be masked because all POs have a possibility to induce mismatch-errors during Scan test.

7.6 HIGH-Z TEST

This test mode is required if the memory chip is packaged together. In High-Z test mode, I/O enable signals are set to high to disable all memory interface I/Os to avoid collision with external test pins for memory test. During High-Z test mode, memory pins can be driven by the external test pins.

Caution: "test_ext_cmosrcv" should be zero or one to set the differential or CMOS receiver I/O.
"test_ext_dfdqs" should be one to set the differential DQS I/O.(LPDDR2, LPDDR3, DDR3)

7.7 NAND TREE TEST

NAND Tree test mode is used to test the input characteristics of I/Os. NAND Tree test will cooperate with the top nand tree test. All I/Os that can be used as input will be set up in the input mode, but the connection of each I/O should be done by Top Integrator. The I/Os that should be tested during NAND Tree test will be DQ[31:0], DQS[3:0].

Caution: "test_ext_cmosrcv" should be zero or one to set the differential or CMOS receiver I/O.
"test_ext_dfdqs" should be one to set the differential DQS I/O.(LPDDR2, LPDDR3, DDR3)
"rst_n" = 0(the clock running of "clk2x" is needed to propagate rst_n=0),
"test_ext_en" = 1'b0
"ctrl_pd[NS:0]" = 'h0. (NS means the number of slice)

8 Application Note

8.1 INITIALIZATION

- After power-up and system PLL locking time, system reset(rst_n) is released.
- Select Memory Type (=PHY_CON0[12:11]).
  - ctrl_ddr_mode=2'b11 (LPDDR3)
  - ctrl_ddr_mode=2'b10 (LDDR2)
  - ctrl_ddr_mode=2'b00 (DDR2)
  - ctrl_ddr_mode=2'b01 (DDR3)

NOTE: If ctrl_ddr_mode[1]=1'b1, lpddr2_cmd=14'h000E(=PHY_CON25[13:0]), cmd_default=14'h000F(=PHY_CON26[13:0])

- Set Read Latency(RL), Burst Length(BL) and Write Latency(WL)
  - Set RL in PHY_CON42[4:0].
  - Set BL in PHY_CON42[12:8].
  - Set WL in PHY_CON26[20:16].
- ZQ Calibration(Please refer to "8.5 ZQ I/O CONTROL PROCEDURE" for more details)
  - Set Drive Strength (=zq_mode_dds or PHY_CON39[27:0]) properly (Please refer to p56, p63).
    - Please don't use default value(=0x0)
  - Enable and Disable "zq_clk_div_en" in PHY_CON16[18]
  - Enable "zq_manual_str" in PHY_CON16[1]
  - Wait until "zq_cal_done"(=PHY_CON17[0]) is enabled.
  - Disable "zq_manual_str"(=PHY_CON16[1])
- Memory Controller should assert "dfi_init_start" from LOW to HIGH.
- Memory Controller should wait until "dfi_init_complete" is set
  - DLL lock will be processed.
  - If the frequency of "clkm" is changed during operation, "ctrl_start" should be clear and set to lock again.
- Enable DQS pull down mode
- Memory Controller should assert "dfi_ctrlupd_req" after "dfi_init_complete" is set.
- Start Memory Initialization.
- Skip the following steps if Leveling and Training are not required.
  - Constraints during Leveling
    - Support BL=4 or 8 during Leveling. (Don't use BL=16)
    - Not support Memory ODT(On-Die-Termination) during Write DQ Calibration.
  - Enable "ctrl_atgate" in PHY_CON0[6].
  - Enable "p0_cmd_en" in PHY_CON0[14].
  - Enable "InitDeskewEn" in PHY_CON2[6].

- Enable "byte_rdlvl_en" in PHY_CON0[13].
- Set "rdlvl_pass_adj=4'h6" in PHY_CON1[19:16].
- Set "ddr3_cmd=14'h105E" as default value (=PHY_CON25[29:16)
- Set "lpddr2_cmd=14'h107E" as default value (=PHY_CON25[13:0])
- Set "cmd_default= 16'h000F(LPDDR2, LPDDR3), 16'h107F(DDR2, DDR3)" as default value (=PHY_CON26[13:0])
- Recommend that "rdlvl_incr_adj=7'h01" for the best margin.
  - Calibration time can be shorter by adjusting "rdlvl_incr_adj" in PHY_CON2[22:16].
- Disable "ctrl_dll_on" in PHY_CON12[5] before Leveling.
  - Read "ctrl_lock_value[8:2]" in PHY_CON13[16:10].
  - Update "ctrl_force[6:0]" in PHY_CON12[14:8] by the value of "ctrl_lock_value[9:2]".
- Write Leveling (refer to 8.1.1)
- CA Calibration(refer to 8.1.2)
- Gate Leveling (refer to 8.1.3)
  - It should be used only for DDR3 (800MHz). Please don't use under 800MHz.
- Read DQ Calibration(=Read Leveling) (refer to 8.1.4)
- After Read DQ Calibration, refer to "T_rddata_en" to know where "dfi_rddata_en_p0/p1" is enabled.
  - Read "T_rddata_en" timing parameters in PHY_CON18 after Read DQ Calibration.
- Write DQ Calibration (refer to 8.1.5)
- Enable "ctrl_dll_on" in PHY_CON12[5].
- Disable "ctrl_atgate" in PHY_CON0[6] if controller controls "ctrl_gate_p0/p1" and "ctrl_read_p0/p1" directly.
- Enable "DLLDeskewEn" (=PHY_CON2[12]) to compensate Voltage, Temperature variation during operation.

Caution: Don't assert "dfi_ctrlupd_req" during Leveling or Training

- Controller should assert "dfi_ctrlupd_req" to make sure All SDLL is updated.
  - If "ca_swap_mode" is enabled, please don't use "ctrl_atgate=1" during normal operation.

NOTE: The goal of data eye training (=Read, Write DQ Calibration) is to identify the delay at which the read DQS rising edge aligns with the beginning and end transitions of the associated DQ data eye. By identifying these delays, the system can calculate the midpoint between the delays and accurately center the read DQS within the DQ data eye.

8.1.1 WRITE LEVELING

Write Leveling compensates for the additional flight time skew delay introduced by the package, board and on-chip with respect to strobe(=DQS) and clock. The flight time skew between DQS and clock should be under 240ps to be compensated properly and suppose that the clock will be delayed than DQS.

- Memory Controller should configure Memory (DDR3, LPDDR3) in Write Level mode.
  - Memory Controller should assert "ODT[1:0]" signals(=dfi_odt_p0/p1) during Write Leveling.
- Configure PHY in Write Level mode.
  - Enable "wrlvl_mode" in PHY_CON0[16].

  - "NOP"(CS HIGH at the clock rising edge N) should be used during "wrlvl_mode"=1(Refer to p105).
- To find out the optimal Write Level De-skew DLL code for the alignment between CK and DQS
  - Set each DLL code (PHY_CON30[6:0], PHY_CON[14:8], PHY_CON[23:17], PHY_CON[30:24]).(1)
    - The start code value should be 0x8. (0x8~0x38)
    - PHY_CON30[6:0] (= "DQS[0]" SDLL code)
    - PHY_CON30[14:8] (= "DQS[1]" SDLL code)
    - PHY_CON30[23:17] (= "DQS[2]" SDLL code)
    - PHY_CON30[30:24] (= "DQS[3]" SDLL code)
  - Update SDLL code(PHY_CON30[16]).(2)
    - Enable "ctrl_wrlvl_resync" in PHY_CON30[16]
    - Disable " ctrl_wrlvl_resync" in PHY_CON30[16]
  - Memory Controller should generate 1 cycle pulse of "dfi_wrdata_en_p0".(3)
  - Memory Controller should read the value of "ctrl_io_rdata[0]" which is output of PHY.(4)
    - If it is zero, Increment "DQS[0]" SDLL code by "1" and then go to "2" to update "DQS[0]" SDLL code.
    - If it is one, the previous "DQS[0]" SDLL code ("The current SDLL code - 1") will be the optimal SDLL code.
  - Memory Controller should read the value of "ctrl_io_rdata[8]" which is output of PHY.(5)
    - If it is zero, Increment "DQS[1]" SDLL code by "1" and then go to "2" to update "DQS[1]" SDLL code.
    - If it is one, the previous "DQS[1]" SDLL code ("The current SDLL code - 1") will be the optimal SDLL code.
  - Memory Controller should read the value of "ctrl_io_rdata[16]" which is output of PHY.(6)
    - If it is zero, Increment "DQS[2]" SDLL code by "1" and then go to "2" to update "DQS[2]" SDLL code.
    - If it is one, the previous "DQS[2]" SDLL code ("The current SDLL code - 1") will be the optimal SDLL code.
  - Memory Controller should read the value of "ctrl_io_rdata[31]" which is output of PHY.(7)
    - If it is zero, Increment "DQS[3]" SDLL code by "1" and then go to "2" to update "DQS[3]" SDLL code.
    - If it is one, the previous "DQS[3]" SDLL code ("The current SDLL code - 1") will be the optimal SDLL code.
- Configure PHY in normal mode after 4~7 are finished.(8)
    - Disable "wrlvl_mode" in PHY_CON0[16].
- Memory Controller should configure Memory (DDR3, LPDDR3) in normal mode.(9)

Caution: Memory Controller should generate 1 cycle pulse of "dfi_wrdata_en_p0".
Memory Controller should add register to read "ctrl_io_rdata[31:0]" if it support Write Leveling.
If the flight time skew between DQS and clock is over 280ps, the skew between DQS and clock can't be compensated by Write Leveling.

8.1.2 CA CALIBRATION

- Controller should configure Memory (LPDDR3) in CA Calibration mode.
- Configure PHY in CA Calibration mode.
  - Enable "wrlvl_mode" in PHY_CON0[16].

  - Enable "ca_cal_mode" in PHY_CON2[23].
- How to find the optimal CA SDLL code. (=PHY_CON10[7:0])
  - Change CA SDLL code in PHY_CON10[7:0]. (1)
    - The start code value should be 0x8.
  - Update CA SDLL code in PHY_CON10[7:0]. (2)
    - Enable "ctrl_resync" in PHY_CON10[24]
    - Disable " ctrl_resync" in PHY_CON10[24]
  - CA to DQ mapping change to calibrate CA[3:0], CA[8:5]. (3)
    - Mode Register Write to MR#41 by Controller.
    - Memory Controller should disable "dfi_cke_p0" and "dfi_cke_p1". (dfi_cke_p0/p1=0)
  - Memory Controller should generate 1 cycle pulse of "dfi_cs_n_p0" with "dfi_address_p0 = 20'h3FF.
  - Memory Controller should read and save the value of "ctrl_io_rddata[15:0]" which is output of PHY.
    - CA[3:0] at rising edge CK(=CA_L[3:0]) is equal to {ctrl_io_rdata[6], ctrl_io_rdata[4], ctrl_io_rdata[2], ctrl_io_rdata[0]}.
    - CA[8:5] at rising edge CK(=CA_L[8:5]) is equal to {ctrl_io_rdata[14], ctrl_io_rdata[12], ctrl_io_rdata[10], ctrl_io_rdata[8]}.
    - CA[3:0] at falling edge CK(=CA_H[3:0]) is equal to {ctrl_io_rdata[7], ctrl_io_rdata[5], ctrl_io_rdata[3], ctrl_io_rdata[1]}.
    - CA[8:5] at falling edge CK(=CA_H[8:5]) is equal to {ctrl_io_rdata[15], ctrl_io_rdata[13], ctrl_io_rdata[11], ctrl_io_rdata[9]}.
  - CA to DQ mapping change to calibrate CA[4], CA[9]. (4)
    - Memory Controller should enable "dfi_cke_p0" and "dfi_cke_p1". (dfi_cke_p0/p1=1)
    - Mode Register Write to MR#48 by Controller.
    - Memory Controller should disable "dfi_cke_p0" and "dfi_cke_p1". (dfi_cke_p0/p1=0)
  - Memory Controller should generate 1 cycle pulse of "dfi_cs_n_p0" with "dfi_address_p0 = 20'h3FF.
  - Memory Controller read and save the value of "ctrl_io_rddata[1:0]" and "ctrl_io_rddata[8:9]" which is output of PHY.
    - CA[4] at rising edge CK(=CA_L[4]) is equal to "ctrl_io_rddata[0]"
    - CA[9] at rising edge CK(=CA_L[9]) is equal to "ctrl_io_rddata[8]"
    - CA[4] at falling edge CK(=CA_H[4]) is equal to "ctrl_io_rddata[1]"
    - CA[9] at falling edge CK(=CA_H[9]) is equal to "ctrl_io_rddata[9]"
  - Check if "CA_L = 10'h3FF" and "CA_H = 10'h000" or not. (5)
  - If not equaled,
    - Go to "6" until it searches for the leftmost code value. (7)
    - If it already saved the leftmost code value, save the current SDLL code by the rightmost code value (=VWMR). Go to "11". (10)
  - If equaled,
    - If it is matched for the first time, save the current SDLL code by the leftmost code value(=VWML). Go to "6". (8)
    - Go to "6" until it searches for the rightmost code value. (9)
  - Increment SDLL code by "1" and then go to "2" to update SDLL code. (6)
- Calculate the optimal CA SDLL code(=PHY_CON10[7:0]). (11)
  - Calculate the optimal CA SDLL code(=VWMC) by the following formula.

        - VWMC = VWML + (VWMR - VWML)/2
    - Update CA SDLL code by using "VWMC".
- Configure PHY in normal mode.
    - Disable "wrlvl_mode" in PHY_CON0[16].
- Memory Controller should configure Memory (LPDDR3) in normal mode.

Caution: Memory Controller should generate 1 cycle pulse of "dif_cs_n_p0".
Memory Controller should add register to read "ctrl_io_rdata[15:0]" if it support CA Calibration.
It is recommended that Memory Controller hold the CA bus stable for one cycle prior to and one cycle after the issuance of MRW CA Training Entry Command to ensure setup and hold timings on the CA bus.

8.1.3 GATE LEVELING

The goal of gate training is to locate the delay at which the initial read DQS rising edge aligns with the rising edge of the read DQS gate(=ctrl_gate_p0/p1). Once this point is identified, the read DQS gate can be adjusted prior to the DQS, to the approximate midpoint of the read DQS preamble. You can use "GATE Leveling" when using "DDR3 memory" over 800MHz.

- Controller should configure Memory (DDR3) in MPR mode. (Please refer to JEDEC Standard.)
- Set "lpddr2_addr=20'h208"(=PHY_CON22[19:0]) to issue MR32 during GATE Leveling (LPDDR2/3)
    - In case of CA swap mode, lpddr2_addr=20'h041 to issue MR32 during GATE Leveling
- Set Gate Leveling Mode.(1)
    - Enable "gate_cal_mode" in PHY_CON2[24]
    - Enable "ctrl_shgate" in PHY_CON0[8]
    - Set "ctrl_gateduradj[3:0] (=PHY_CON1[23:20]) in the following way.
        - 4'b0000" (DDR3, DDR2)
        - 4'b1011" (LPDDR3)
        - 4'b1001" (LPDDR2)

Warning: Don’t use Gate Leveling for productions in case of LPDDR2 or LPDDR3.

- Memory Controller should assert "dfi_rdlvl_en" and "dfi_rdlvl_gate_en" to do read leveling.(2)
- Memory Controller should wait until "dfi_rdlvl_resp" is set.(3)
    - The maximum waiting time will be 20us. If the any command (the refresh or pre-charge command) is required within 20us, please issue those commands before "(1)".
- Memory Controller should deassert "dfi_rdlvl_en" and "dfi_rdlvl_gate_en" after "dfi_rdlvl_resp" is disabled.
- Disable DQS pull down mode.(4)
- Memory Controller should configure Memory (DDR3) in normal mode.

8.1.4 READ DQ CALIBRATION (=READ LEVELING, READ DESKEWING)

Read DQ Calibration adjusts for the delays introduced by the package, board and on-chip that impact the read cycle.

- In case of using DDR3 Memory,
  - Memory Controller should configure Memory in MPR mode. (Please refer to JEDEC Standard)
  - Set "PHY_CON1[15:0]" by "0xFF00" if "Pre-defined Data Pattern" is "[0x0000_0000,0x0101_0101, 0x0000_0000,0x0101_0101]" (byte_rdlvl_en=1)
  - Set "PHY_CON1[15:0]" by "0x0100" if "Pre-defined Data Pattern" is "[0x0000_0000,0xFFFF_FFFF, 0x0000_0000,0xFFFF_FFFF]" in MPR mode. (byte_rdlvl_en=1)

NOTE: Read DQ Calibration with "byte_rdlvl_en=0" should be done after Write DQ Calibration.
Set "PHY_CON1[15:0]" by "0xFF00". "MPR Data Pattern" should be "[0x0000_0000,0xFFFF_FFFF, 0x0000_0000,0xFFFF_FFFF]".

- In case of using LPDDR3 or LPDDR2 Memory,
  - Set "PHY_CON1[15:0]" by "0x00FF" if "MRR32 DQ Pattern" is "[0x0101_0101,0x0000_0000, 0x0101_0101, 0x0000_0000]" (byte_rdlvl_en=1)
  - Set "PHY_CON1[15:0]" by "0x0001" if "MRR32 DQ Pattern" is "[0xFFFF_FFFF,0x0000_0000, 0xFFFF_FFFF, 0x0000_0000]". (byte_rdlvl_en=1)

NOTE: Read DQ Calibration with "byte_rdlvl_en=0" should be done after Write DQ Calibration.
Set "PHY_CON1[15:0]" by "0x00FF". "MRR32 DQ Pattern" should be "[0xFFFF_FFFF,0x0000_0000, 0xFFFF_FFFF, 0x0000_0000]".

  - Set "lpddr2_addr=20'h208"(=PHY_CON22[19:0]) to issue MR32 during Calibration (LPDDR2/3)
    - In case of CA swap mode, lpddr2_addr=20'h041 to issue MR32 during Calibration.
- Set Read Leveling Mode.(1)
  - Enable "rd_cal_mode" in PHY_CON2[25]
  - ~~Calibration time can be shorter by adjusting "rdlvl_incr_adj" in PHY_CON2[22:16].~~
    - ~~Recommend that PHY_CON2[22:16] = 7'b000_0001 for the best margin.~~
- Memory Controller should assert "dfi_rdlvl_en" to do read leveling.(2)
- Memory Controller should wait until "dfi_rdlvl_resp" is set.(3)
  - The maximum waiting time will be 20us. If the any command (the refresh or pre-charge command) is required within 20us, please issue those commands before "(1)".
- Memory Controller should deassert "dfi_rdlvl_en" after "dfi_rdlvl_resp" is enabled.(4)
- Memory Controller should configure Memory (DDR3) in normal mode.

8.1.5 WRITE DQ CALIBRATION (=WRITE DESKEWING)

Write DQ Calibration adjusts for the delays introduced by the package, board and on-chip that impact the write cycle.

- Set Write Latency(WL) before Write Training(1)
  - Set "T_wrdata_en" by "WL" in PHY_CON26[20:16].
    - WL is defined as clock cycles between the write command and the first valid rising edge of DQS
- Memory Controller should issue "Active Command".
- PHY will keep writing and reading the pattern in "PHY_CON1[15:0]" for Write DQ Calibration according to the following settings.(2)
  - The column address should be defined in "lpddr2_addr" in PHY_CON22[19:0](LPDDR2, LPDDR3).
    - "lpddr2_addr[9:0]" correspond to CA[9:0] at the rising edge of CK, and "lpddr2_addr[19:10]" to CA[19:10] at the falling edge of CK.

        - It should be the "READ" command. For example, lpddr2_addr=20'h5 if the column address is 11'h0 and bank address is 3'b000. In case of CA swap mode, lpddr2_addr=20'h204 if the column address is 11'h0 and bank address is 3'b000.
    - The column address should be defined in "ddr3_addr" in PHY_CON24[15:0] (DDR3)
        - For example, if the column address (=ddr3_addr) is "0x0", PHY will write and read the pre-defined pattern (=PHY_CON1[15:0]) at 0x0, 0x4, 0x8 and 0xC for DQ Calibration. (BL=4)
    - In case of using LPDDR2 or LPDDR3 memory,
        - Set "PHY_CON1[15:0]=0x0001" and "byte_rdlvl_en=1"(=PHY_CON0[13])
        - Set "PHY_CON1[15:0]=0x00FF" and "byte_rdlvl_en=0"(=PHY_CON0[13]) for Deskewing.
    - In case of using DDR3 memory,
        - Set "PHY_CON1[15:0]=0x0100" and "byte_rdlvl_en=1"(=PHY_CON0[13])
        - Set "PHY_CON1[15:0]=0xFF00" and "byte_rdlvl_en=0"(=PHY_CON0[13]) for Deskewing.
- Enable "p0_cmd_en" in PHY_CON[14].
- Set Write Training Mode(3)
    - Enable "wr_cal_mode" in PHY_CON2[26].
- Enable "wr_cal_start" in PHY_CON2[27] to do Write DQ Calibration.(4)
- Memory Controller should wait until "dfi_rdlvl_resp" is set.(5)
    - The maximum waiting time will be 50us. If the any command (the refresh or pre-charge command) is required within 50us, please issue those commands before "(3)".
- Disable "wr_cal_start" in PHY_CON2[27] after "dfi_rdlvl_resp" is enabled.(6)

8.2 LOW FREQUENCY OPERATION

Even if the operation frequency of "clk2x" is out of the range of MDLL Input frequency(400~800MHz), DDR PHY can operate at the low frequency because MDLL Input clock is separated from PHY Input clock. It is recommended that Read Leveling should be done in the normal operation. The following sequence is how to operate PHY at low frequency with the different MDLL clock

- If "dfi_init_start" = 0, Controller should assert "dfi_init_start" from LOW to HIGH.
- After "dfi_init_complete" is checked by controller, read the value of "ctrl_lock_value".
- Enter Self-Refresh(CKE=0)
- Go to the low frequency.
    - Change "clk2x" to the low frequency, but don't change the frequency of "clkm"(400~800MHz).
- Write the multiplied value of ctrl_lock_value[8:2] to ctrl_force[6:0].
    - For example, if the operation frequency will be the half of MDLL clock, two multiplied value of ctrl_lock_value should be written to "ctrl_force".
    - If "the multiplied value" is more than 0x7F, "ctrl_force" will be "0x7F".
- the low frequency is under 100MHz,
    - "ctrl_force"=0x7F, "ctrl_offsetd"=0x7F, ctrl_offsetr*=0x7F, ctrl_offsetw*=0x7F(* means 0~3)
    - CA0DeSkewCode ~ CA9DeSkewCode = 0x60.
    - Turn off "ctrl_dll_on".
- "dfi_ctrlupd_req" should be issued more than 10 cycles after ctrl_dll_on is disabled.
- Exit Self-Refresh(CKE=1).

- Operate in the low frequency.

Caution: "ctrl_atgate"(=PHY_CON0[6]) should be "0" under 400MHz(=clk2x). "ctrl_gate_p0/p1" and "ctrl_read_p0/p1" should be generated from controller.

If it goes back to the original high frequency again, please refer to the following procedures.

- Enter Self-Refresh(CKE=0)
- If "ctrl_dll_on=0", turn on "ctrl_dll_on".
- "ctrl_offsetd"=0x08, ctrl_offsetr*=0x08, ctrl_offsetw*=0x08(* means 0~3)
- CA0DeSkewCode ~ CA9DeSkewCode = 0x08.
- Wait until "ctrl_clock=1".
- Controller should assert "dfi_ctrlupd_req" to apply the new "ctrl_lock_value" after "ctrl_clock=1".
- Exit Self-Refresh(CKE=1)

8.3 OFFSET CONTROL

ctrl_offset0~3 control the offset of 90° phase shift of DQS or 270° clock. ctrl_offsetd and ctrl_offset0~ctrl_offset3 are just used for debug or margin test purpose (after that, it is possible to program offset for the compensation to maximize the margin).

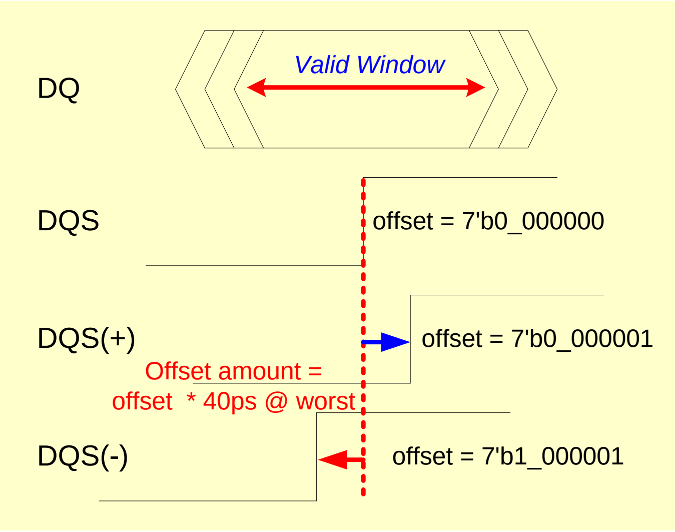

Figure 8-1. Offset Control Example for ctrl_offset0~ctrl_offset3

8.4 DLL LOCK PROCEDURE

- After power-up and system PLL locking time, system reset(rst_n) is released.
- Assert "dfi_init_start" from LOW to HIGH.
- When DLL lock is finished, "dfi_init_complete" is set.
- Before memory access, "dfi_ctrlupd_req" should be applied. It's recommended that "dfi_ctrlupd_req" should be set and clear at the start of refresh cycle automatically by the memory controller to update DLL lock information periodically.

DLL is used to compensate PVT condition. Therefore DLL should not be turned-off for reliable operation except for the case of frequency scaling.(only to lower frequency scaling is permitted)

To turn-off the DLL, follow the next steps.

- After DLL locking, CPU reads ctrl_lock_value and write ctrl_lock_value[9:2] to ctrl_force.
- DLL can be turned off by clearing ctrl_dll_on.
- "dfi_ctrlupd_req" should be issued 6 cycles after ctrl_dll_on was set and cleared.

8.5 ZQ I/O CONTROL PROCEDURE

ZQ I/O calibrates the I/Os to match the driving and termination impedance by referencing resistor value of resistor(RZQ) connected externally from ZQ pin to ground. For DDR2/DDR3, RZQ should be 240ohm. One-time calibration is provided for ZQ I/O control procedure. There are two modes for one-time calibration. One is “Long calibration mode” and the other is “Short calibration mode”.

8.5.1 One-time calibration

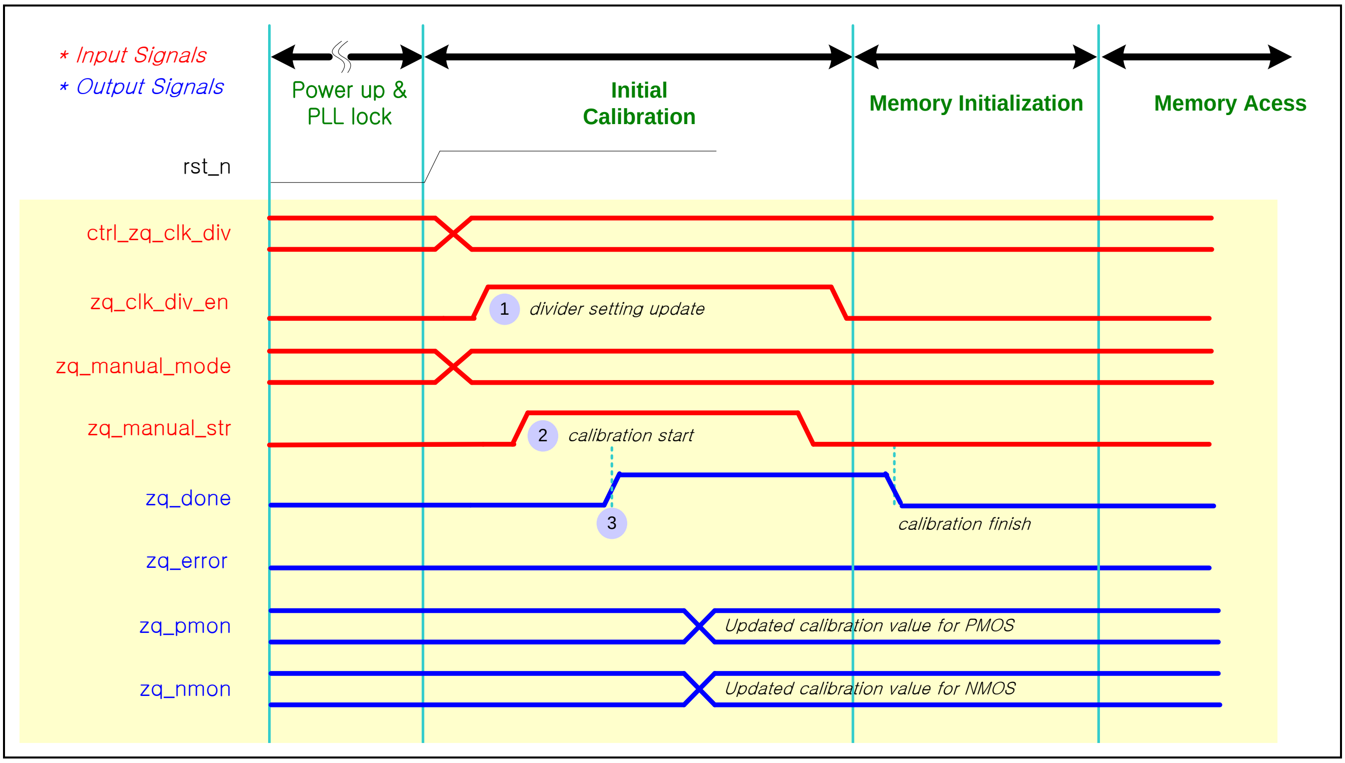

Figure 8-2. One-time calibration procedure

- After power-up and system PLL locking time, system reset(rst_n) is released.
- Set ctrl_zq_clk_div[31:0] to proper value(=0x7)
- Set zq_clk_div_en from 1’b0 to 1’b1 to update divider settings (ctrl_zq_clk_div[31:0]=0x7).
- Set zq_manual_mode
  - Long calibration mode: 2’b01
  - Short calibration mode: 2’b10
- Start ZQ I/O calibration by setting zq_manual_str from 1’b0 to 1’b1
- When calibration is done, zq_done(=PHY_CON17[0]) will be set. ~~for four cycles (system clock)~~
- After zq_done(=PHY_CON17[0]) is asserted, clear zq_manual_str.
- Clear zq_clk_div_en

8.5.2 Manual setting

- After power-up and system PLL locking time, system reset(rst_n) is released.
- Set ctrl_zq_clk_div[31:0] to proper value(=0x7)
- Set zq_clk_div_en from 1’b0 to 1’b1 to update divider settings(ctrl_zq_clk_div[31:0]=0x7).
- Set zq_manual_mode=2'b00(=Force calibration mode).
- Start ZQ I/O calibration by setting zq_manual_str from 1’b0 to 1’b1
- After zq_done(=PHY_CON17[0]) is asserted, clear zq_manual_str

8.6 TERMINATION RESISTOR CONTROL

8.6.1 Termination Control for Write

During data write, termination resistor will be turned-on to eliminate reflection and is controlled by ODT signal. ODT pin should be driven HIGH during data write. (Please refer to the memory specification for more information)

In this example (Figure 8-4), suppose that "tAOND" = 2cycles and "tAOFD" = 2.5 cycles. dfi_odt_p0/p1 will control external ODT pin.

- dfi_wrdata_en_p0/p1(active HIGH) should be driven after " WL-2" from command issue during "(BL/2)" cycles.
- dfi_wrdata_p0/p1,dfi_dm_p0/p1(valid value) should be driven after "WL" cycles from command issue during "(BL/2)" cycles.
- According to the value of "tAOND" and "tAOFD", dfi_odt_p0/p1 should be driven after "WL-3" cycles from command issue during "(BL/2)+1" cycles. (Please refer to "tAOND" and "tAOFD" in the memory specification).
  - For the case of DDR3, ODT will be driven with command issue during the minimum "(BL/2)+2" cycles. (Please refer to the memory specification for the more information)

NOTE: WL(=Write Latency), BL(=Burst Length)

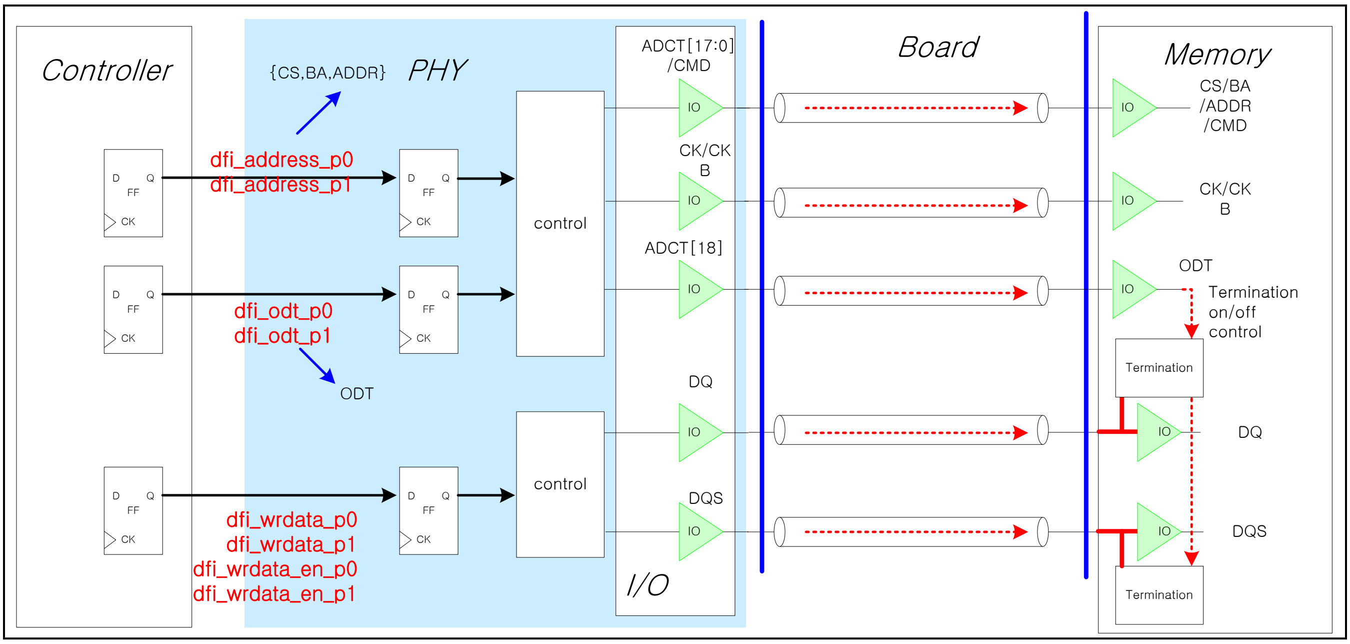

Figure 8-3. Data Write Path and Assignment Example

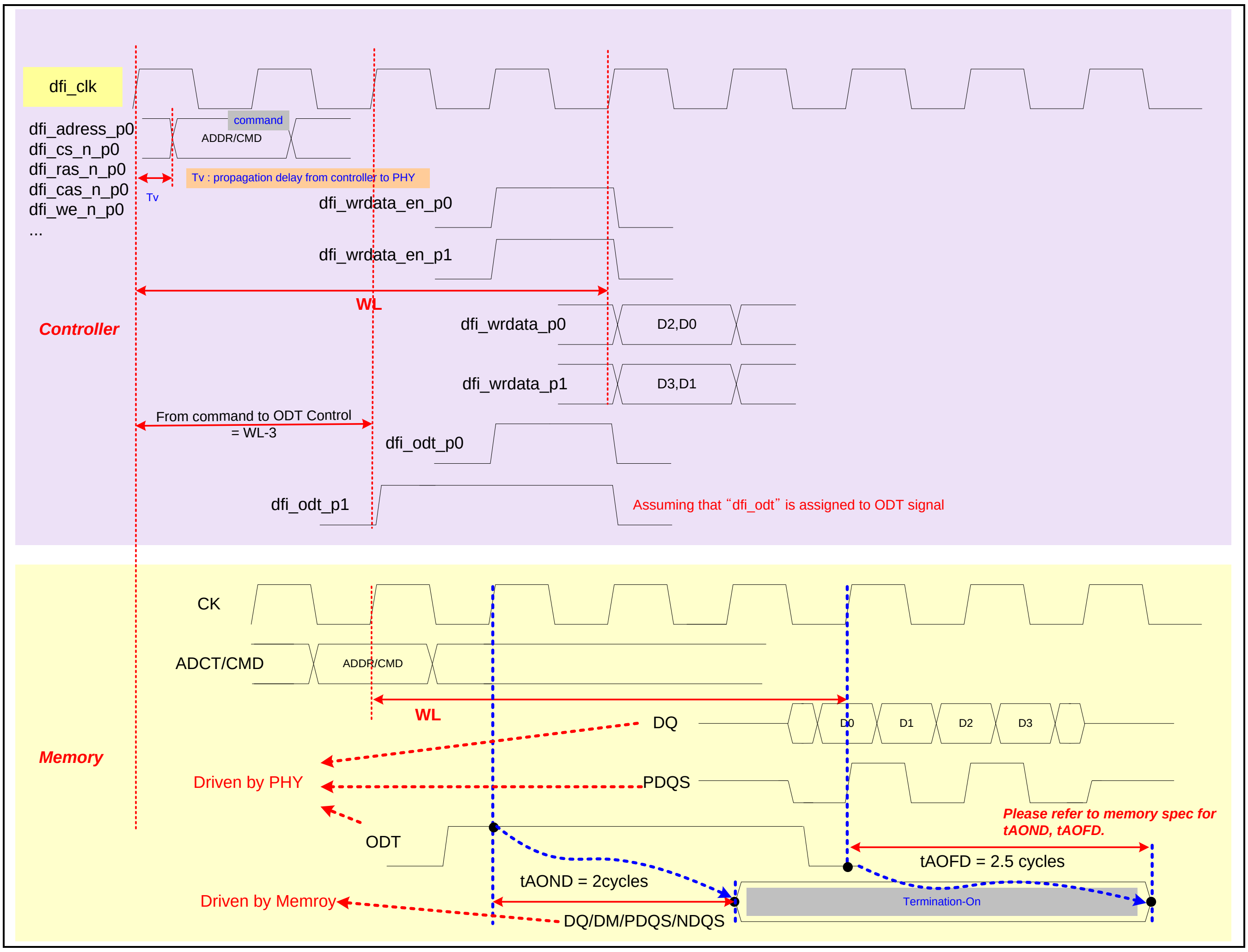

Figure 8-4. Timing Diagram to Control ODT

8.6.2 Termination Control for Read

During data read, termination resistor should be turned-on to eliminate reflection and is controlled by ctrl_read_p0/p1 signal from controller. If ctrl_atgate=1, PHY can internally generate "ctrl_read_p0/p1".

NOTE: "ctrl_readadj" can control the time on a cycle base when termination register is turned-on.
"ctrl_readduradj" can control the duration on a cycle base that termination register is truned-on.
"ctrl_read_width" can control the time on half cycle when termination register is turned-on.(Please refer to p41)

If ctrl_gate=0, ctrl_read_p0/p1 can be driven HIGH after "(RL-1)/2" or "(RL-1)/2 + 1" cycles from command issue from controller and driven LOW after "[RL + (Burst Length / 2)]/2" or "[RL + (Burst Length / 2)]/2 + 1" cycles from command issue. (Please refer to Figure 8-6) The duration of ctrl_read_p* "HIGH" will be "(Burst Length / 2) + 1". When termination resistor is used during read operation, DDR PHY always requires to add an idle cycle to "tRTW" (For example, tRTW=5 for BL4, tRTW=7 for BL8 in case of DDR3). Be careful that this requirement is different with memory specification (Please refer to JEDEC).

* RL (Read Latency) = AL(Additive Latency) + CL(CAS Latency)

Caution: If there is no idle cycle after read, the following write can be failed due to turning on the termination register. It is strongly recommended that the memory controller should provide programmable options to add more idle cycles after read.

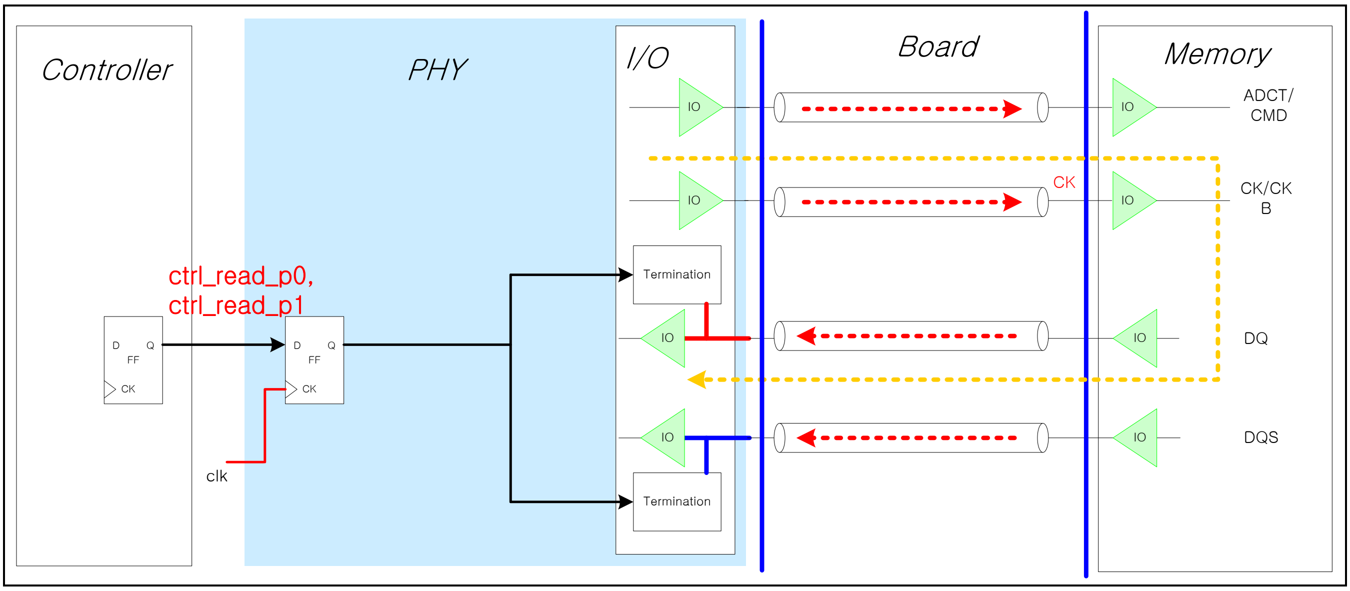

Figure 8-5. Data Read Path

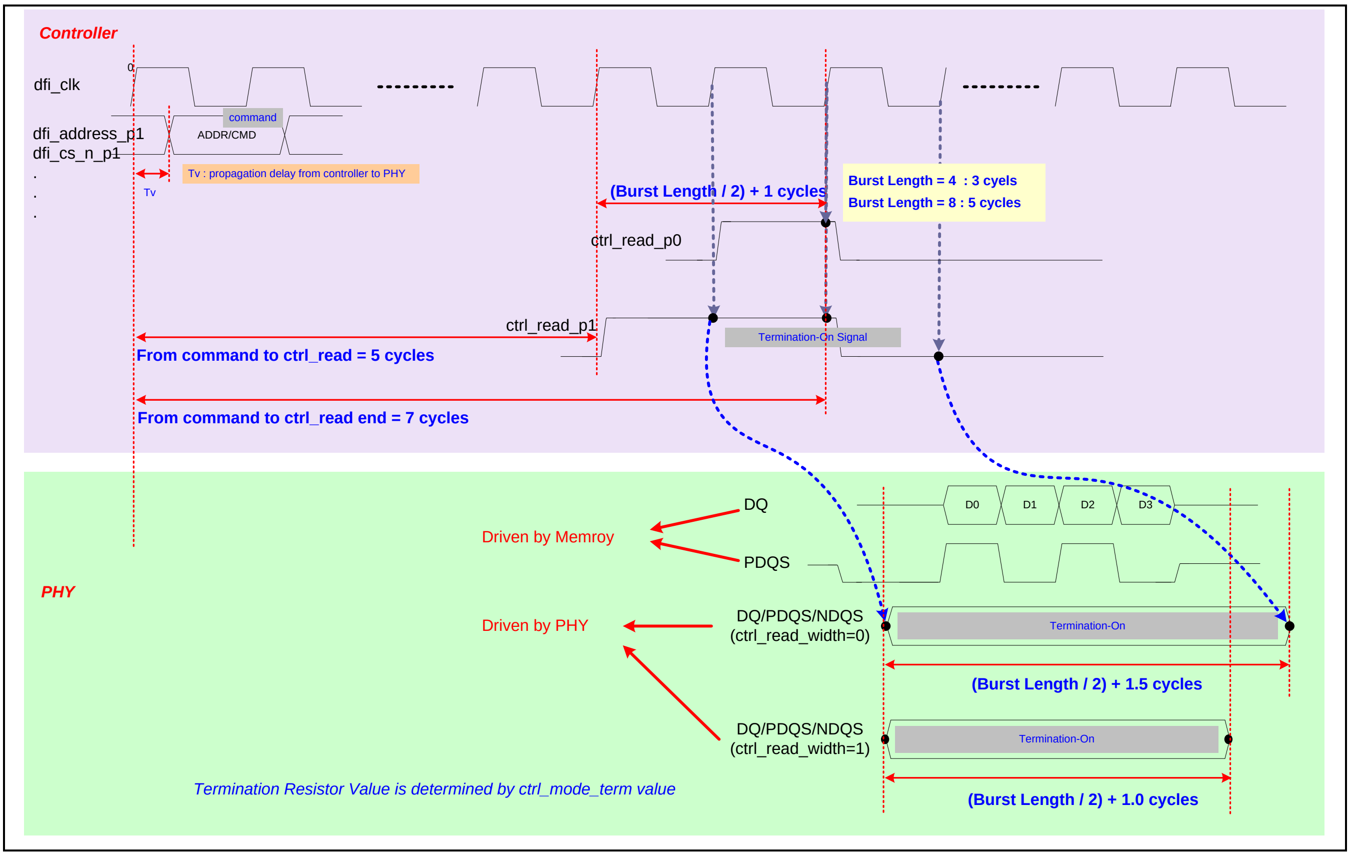
Controller
dfi_clk
α
dfi_address_p1
dfi_cs_n_p1
command
ADDR/CMD
Tv : propagation delay from controller to PHY
Tv
(Burst Length / 2) + 1 cycles
Burst Length = 4 : 3 cyels
Burst Length = 8 : 5 cycles
ctrl_read_p0
ctrl_read_p1
Termination-On Signal
From command to ctrl_read = 5 cycles
From command to ctrl_read end = 7 cycles
PHY
DQ
D0
D1
D2
D3
Driven by Memroy
PDQS
Driven by PHY
DQ/PDQS/NDQS
(ctrl_read_width=0)
Termination-On
(Burst Length / 2) + 1.5 cycles
DQ/PDQS/NDQS
(ctrl_read_width=1)
Termination-On
Termination Resistor Value is determined by ctrl_mode_term value
(Burst Length / 2) + 1.0 cycles

Figure 8-6. Timing Diagram to Control ctrl_read_p0/p1 (Phase1 Read, RL=11)

8.7 DLL CODE UPDATE

Assertion of the dfi_ctrlupd_req signal indicates the control, read and write interfaces on the DFI are idle. While the dfi_ctrlupd_ack signal is asserted, the DFI bus may only be used for commands related to the update process.

The MC guarantees that dfi_ctrlupd_req signal will be asserted for at least tctrlupd_min cycles, allowing the PHY time to respond. To acknowledge the request, the dfi_ctrlupd_ack signal must be asserted while the dfi_ctrlupd_req signal is asserted. The dfi_ctrlupd_ack signal must de-assert at least one cycle before tctrlupd_max expires. (Tctrlupd_min=2, Tctrlupd_max=20)

In case of LPDDR2/LPDDR3, The PHY is expected to drive values on dfi_address[9:0] to ADCT[9:0] on the rising edge of clock and value on dfi_address[19:10] to ADCT[9:0] on falling edge of memory clock. But PHY will drive value on dfi_address[19:10] of controller to ADCT[9:0] during 6 clock cycles after "dfi_ctrlupd_req" was issued. So controller should use NOP command with CSN=1 during the refresh period or during ctrl_wake_up=0.

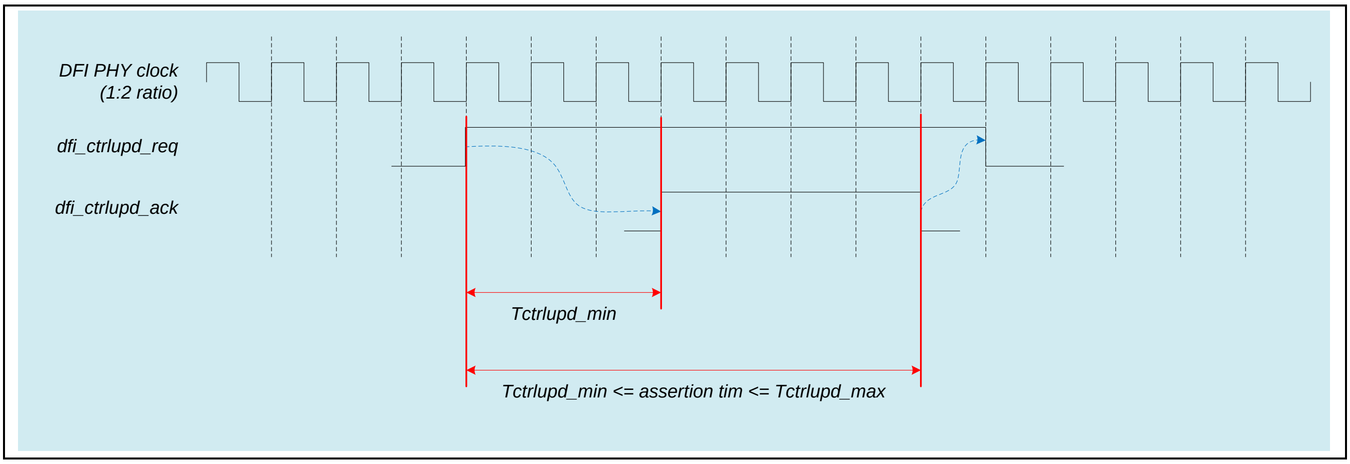

Figure 8-7. MC-Initiated Update Timing Diagram

When CPU or MC change the code value in SDLL or De-skew DLL by using APB Interface, "ctrl_resync" (PHY_CON[24]) or "wrlvl_resync"(PHY_CON30[16]) should be high and then low to apply the updated code values. And the following conditions should be met during the assertion of "dfi_ctrlupd_req", "ctrl_resync" or "wrlvl_resync", (Please refer to p90)

- CS=1 or CKE=0 if any CA SDLL(PHY_CON24) or DeSkew(PHY_CON30) Code is changed.
- Any Write operation shouldn't be permitted if Write SDLL(PHY_CON6) Code is changed.

Caution: "NOP"(CS HIGH at the clock rising edge) should be used during the assertion of "dfi_ctrlupd_req", "ctrl_resync" or "wrlvl_resync". Don't use "NOP"(CS LOW at the clock rising edge).

Caution: If the DeSkew Code for CKE[1:0], CS[1:0], CK, RESET is changed, CKE should be always "LOW" during the assertion of "dfi_ctrlupd_req", "ctrl_resync" or "wrlvl_resync".

8.8 CLOCK CONTROL

dfi_dram_clk_disable controls CK/CKB signals.

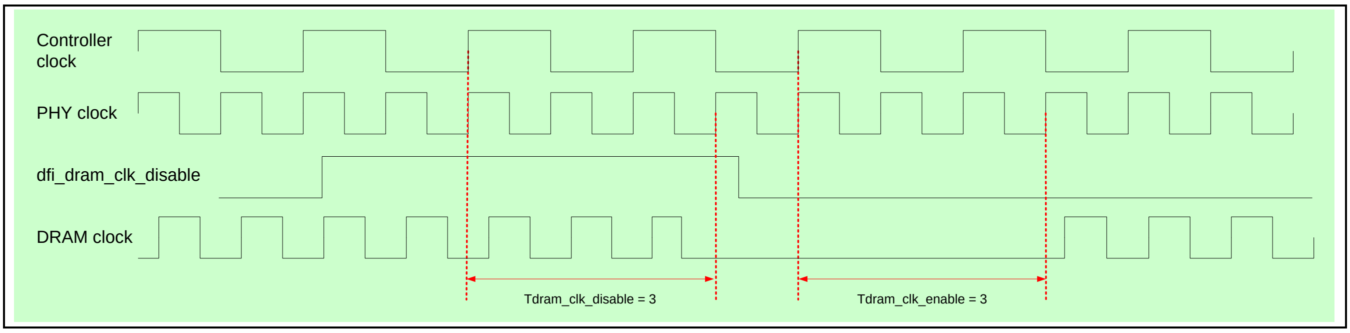

Figure 8-8. Clock Control Timing

8.9 SLAVE DLL Disabling

"ctrl_wake_up" can block the input clock of SDLL during idle period on cycle base. Memory Controller can reduce a lot of DLL power by controlling "ctrl_wake_up" properly. "ctrl_wake_up" in ctrl_slice can control SDLL for CA(LPDDR2/3). And "ctrl_wake_up" in data_slice can control SDLL for DQ(Write). If LPDDR2 is not used, PHY won't use SDLL in ctrl_slice to generate command signals. So "ctrl_wake_up" in ctrl_slice can be always disabled.

"ctrl_wake_up" interface timing for read is showed in Figure 8-9. If there is any read command in Figure 8-9, "ctrl_wake_up" for ctrl_slice should be issued 2 cycles before any command is activated. But "ctrl_wake_up" for data_slice don't need to be enabled because there is no write data which has to be transferred to memory.

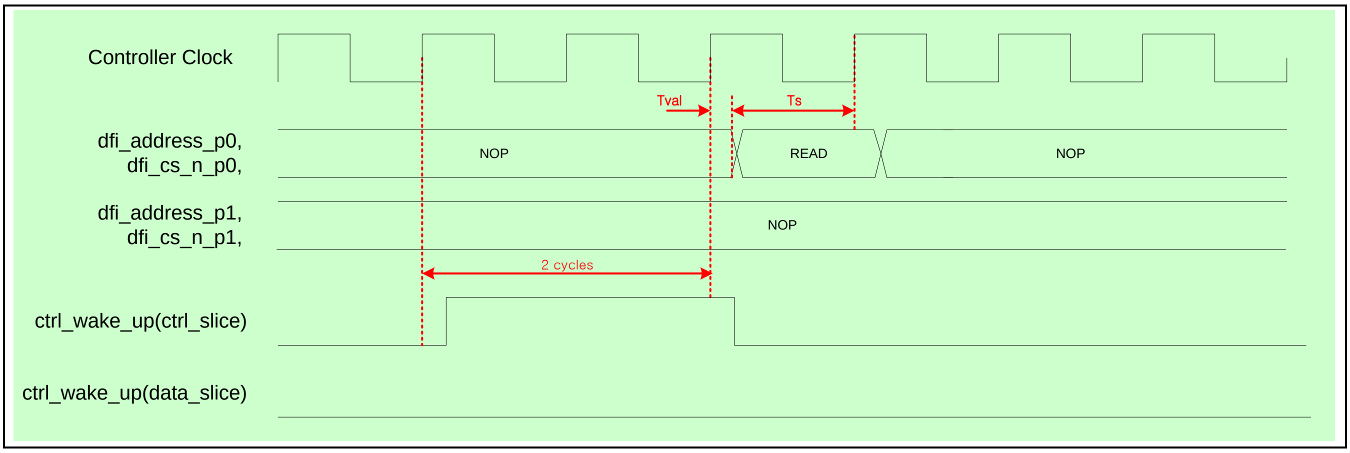

Figure 8-9. SLAVE DLL Control for Read(phase0)

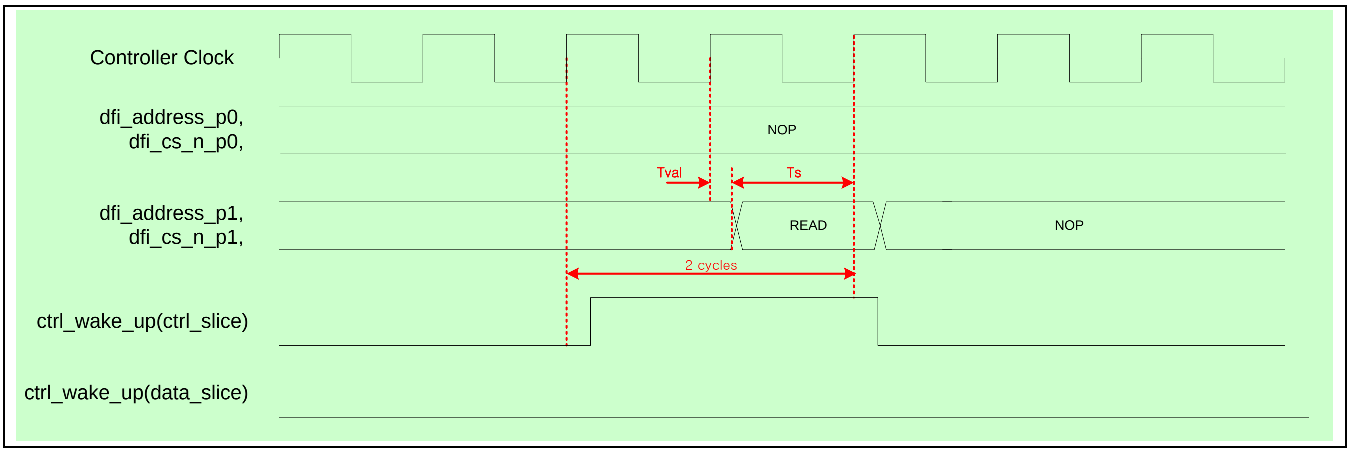

Figure 8-10. SLAVE DLL Control for Read(phase1)

"ctrl_wake_up" interface timing for write is showed in Figure 8-11. If there is any write command as Figure 8-11, "ctrl_wake_up should be enabled when "dfi_wrdata"_en is issued and it can be disabled 1 cycle after "dfi_wrdata_en" is deasserted. If any command requires "data_slice" to be used, you should follow Figure 8-11, but if not, you can follow Figure 8-9.

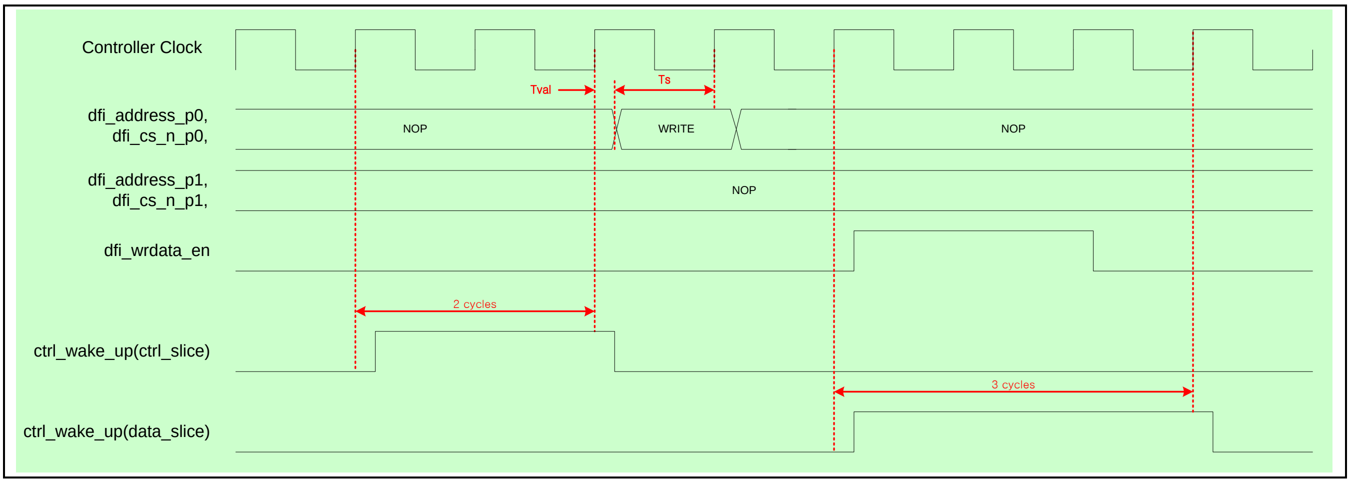

Figure 8-11. SLAVE DLL Control for Write(phase0)

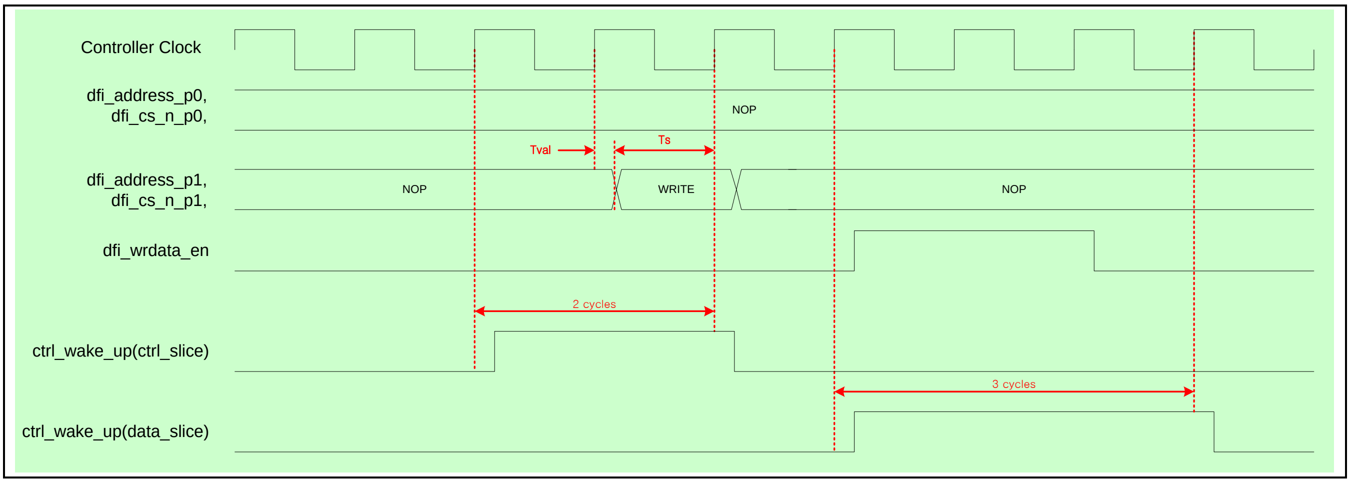

Figure 8-12. SLAVE DLL Control for Write(phase1)