DDR3 HPHY

DDR3 PHY (V6R0)

Revision 1.22

August 2013

DataSheet

SAMSUNG ELECTRONICS

SAMSUNG

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.

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

Contact Us: *suhoange@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	2012-02-15	• Preliminary Spec	S.H. Kim
0.20	2012-04-07	• Correct "PHY Block Diagram"(p20) • Remove "ddr3_cmd"(p68) • Change bit-width of "ctrl_force"(p71) • Add "Calibration Status register"(p76) • Add "Valid Window Margin register"(p77) • Add "Read Deskew register"(p83) • Add "Write Deskew register"(p88) • Correct "S/W Write Leveling"(p140) • Add "H/W Write Leveling"(p140) • Correct "DLL Code Update"(p156)	S.H. Kim
0.30	2012-05-11	• Add " ctrl_place_type[9:0]"(p27) • Correct "ctrl_upd_range"(p51) • Remove "FastDeskewStart"(p53)	S.H. Kim
0.40	2012-05-18	• Correct "DIRECT ACCESS INTERFACE SIGNALS"(p39) • Add "ctrl_wr_dis" bit field(p51) • Add Note(p55) • Add CAL_WL_STAT register(p76) • Add "CTRL_IO_RDATA0/1/2" register(p109) • Add "Cautions"(p137,138)	S.H. Kim
0.60	2012-07-05	• Add "RODT_CON0" Register(p57) • Change "upd_mode" reset value in "OFFSETD_CON0" register(p66) • Add "ZQ I/O" Test mode(p126) • Correct description in I/O test (p44, p137) • Add "upd_mode" setting sequence(p139) • Correct procedures for S/W Write Leveling(p141) • Remove the setting of "wl_cal_mode"(p145) • Emphasize requirement in red during DLL Update(p156) • Correct Timing Diagram(p158)	S.H. Kim H.K. Lee
0.70	2012-07-11	• Correct ZQ Control Procedure(p150)	S.H.Kim
0.80	2012-08-07	• Add description about clkm(p27) • Add Caution about Reset(p130) • Correct "rdlvl_pass_adj=4"(p139) • Add " ctrl_readduradj=1"(p140) • Correct typo about "dfi_odt_p0/p1"(p141) • Remove the setting procedure about ctrl_offset*(p149)	S.H.Kim
0.90	2012-12-17	• Add cautions when setting DQS pull down(p139) • Correct description about ctrl_read_*(p154)	S.H.Kim

		• Correct Figure 8-6(p155)	
1.00	2012-12-26	• Correct setting for "update_mode"(p139) • Correct description(p146)	S.H.Kim
1.10	2013-01-03	• Add setting of "reg_mode[0]=1'b1" in case of H/W Write leveling(p141)	S.H.Kim
1.20	2012-01-16	• Correct description about "reg_mode[7:0]"(p54) • Remove the manual setting of *_DESKEW_CON*(p87,93) • Correct typo about "zq_mode_term", 3'b100 → 3'b001(p73) • Add the setting of "ctrl_resync"(p141) • Correct "wl_cal_mode" and "wl_cal_start"(p145)	S.H.Kim
1.21	2013-03-19	• Correct the description about "clkm duty requirement"(p27) • Correct the setting value of "ctrl_place_type[*]"(p27) • Remove "ctrl_ckdis"(p51) • Correct Figure 5-1(p111) • Add Figure 5-9(p118) • Correct Figure 5-12(p121) • Correct Figure 5-13(p122) • Correct default value of "test_ext_offsetr", 8'h0 → 8'h8(p128) • Remove "test_ext_rdlvl_vwml, test_ext_rdlvl_vwmr"(p128) • Improve Figure 7-3(p132) • Improve Figure 7-4(p133) • Improve Figure 7-6(p135) • Add caution about changing the frequency or voltage(p139) • Add Write Leveling DLL manual setting guide(p142) • Remove "test_ext_dll_on"(p130)	S.H.Kim
1.22		• Add DDS setting information for CaAdrDrvrDS(p70) • Add Timing between RESET and clk2x(p114)	S.H.Kim S.Y.YI

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

Table Number | **Title** | **Page Number**

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
Slice	DDR PHY consists of multiple slices. Each slice will be connected to 8-bit DQs, 1-bit DM and 1-bit DQS. For example, 32bit PHY has 4 slices (=data_slice). slice0 is connected to DQ[7:0], DM[0] and DQS[0], slice1 is connected to DQ[15:8], DM[1] and DQS[1], slice2 is connected to DQ[23:16], DM[2] and DQS[2], slice3 is connected to DQ[31:24], DM[3] and DQS[3].
DS#	Data Slice, # means 0, 1, ~ 8. For example, DS0 means "data_slice 0"
NS	The number of "Data Slice". If using 32bit PHY, NS will be 4.

List of Acronyms

Acronyms	Descriptions

1 Overview

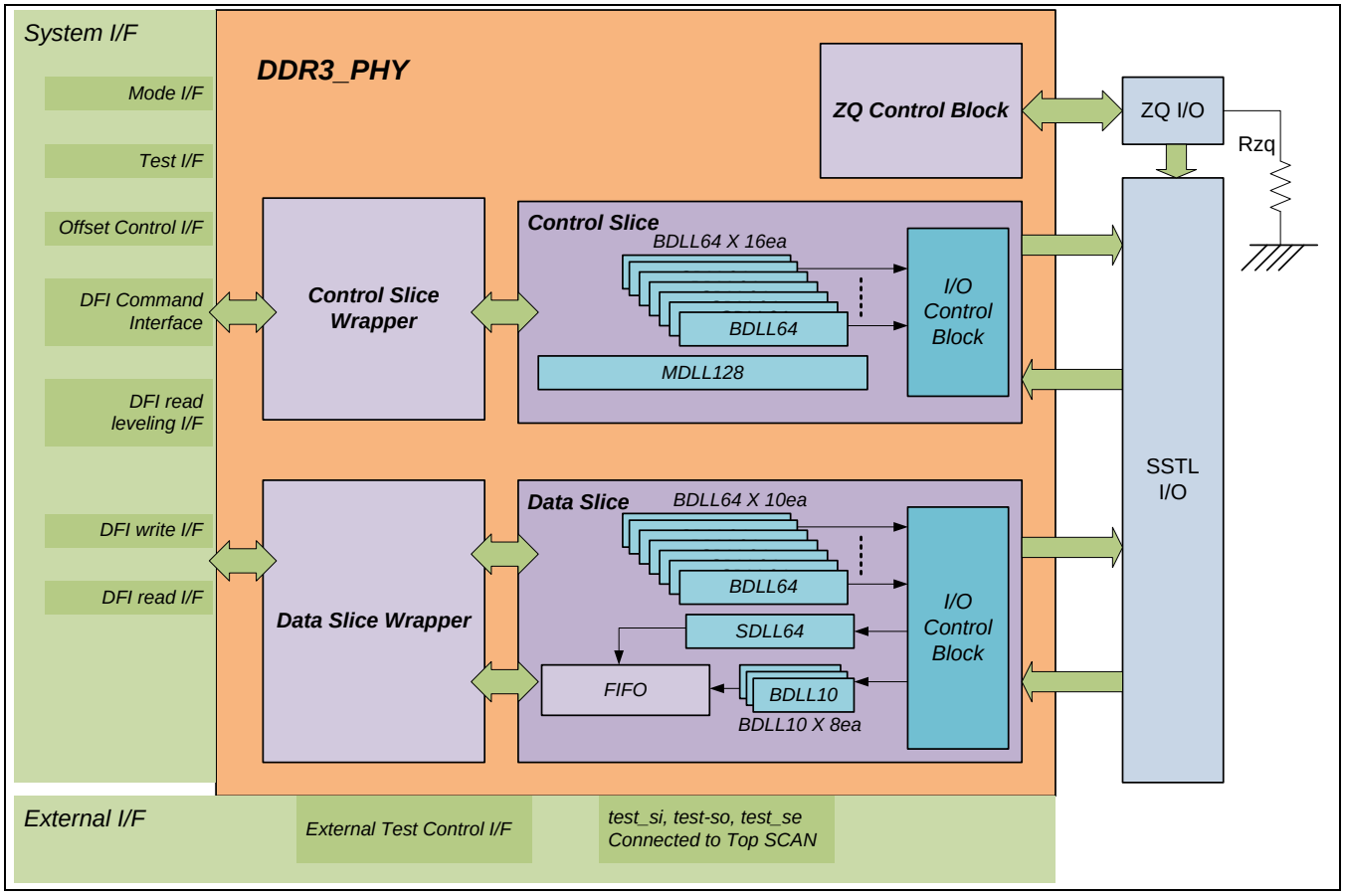

Figure 1-1 LPDDR3 HPHY

The DDR3 PHY(V6R0) provides the following features:

- DDR3 is 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.
- Write Leveling, Gate Leveling, Read Training and Write Training are supported.
- 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.

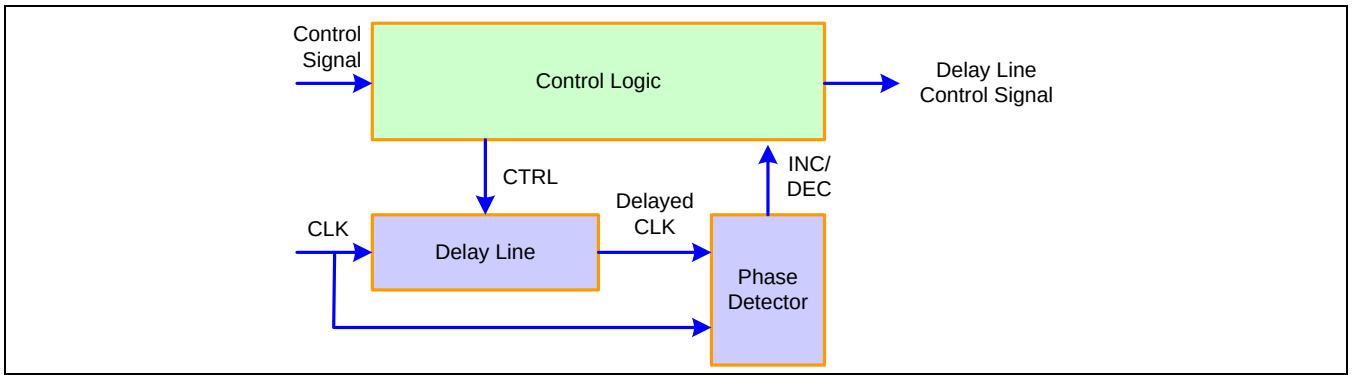

Figure 2-1. Block Diagram of DLL

2.1.2 Control

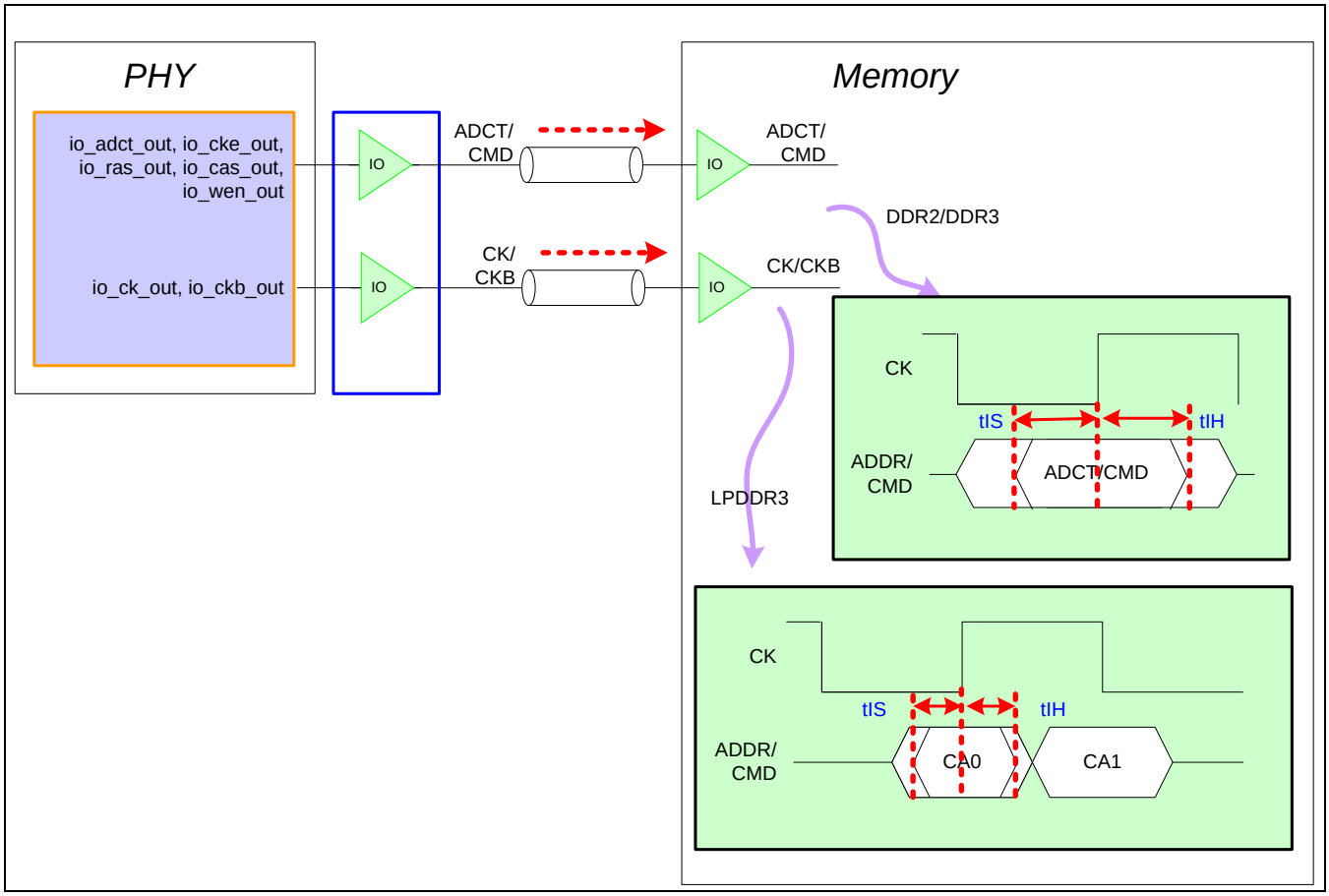

Figure 2-2. Block Diagram of the Control Path

Control block generates address, control and clock signals to I/Os. ADCT[24:0] are reset(RSN)/ODT/address (ADDR)/bank(BA)/chip-select(CS) signals and RAS, CAS, WE are command signals. GATEO and GATEI are DQS clean 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.

GATEO and GATEI signals are used to clean strobe signal, i.e. DQS cleaning. 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.

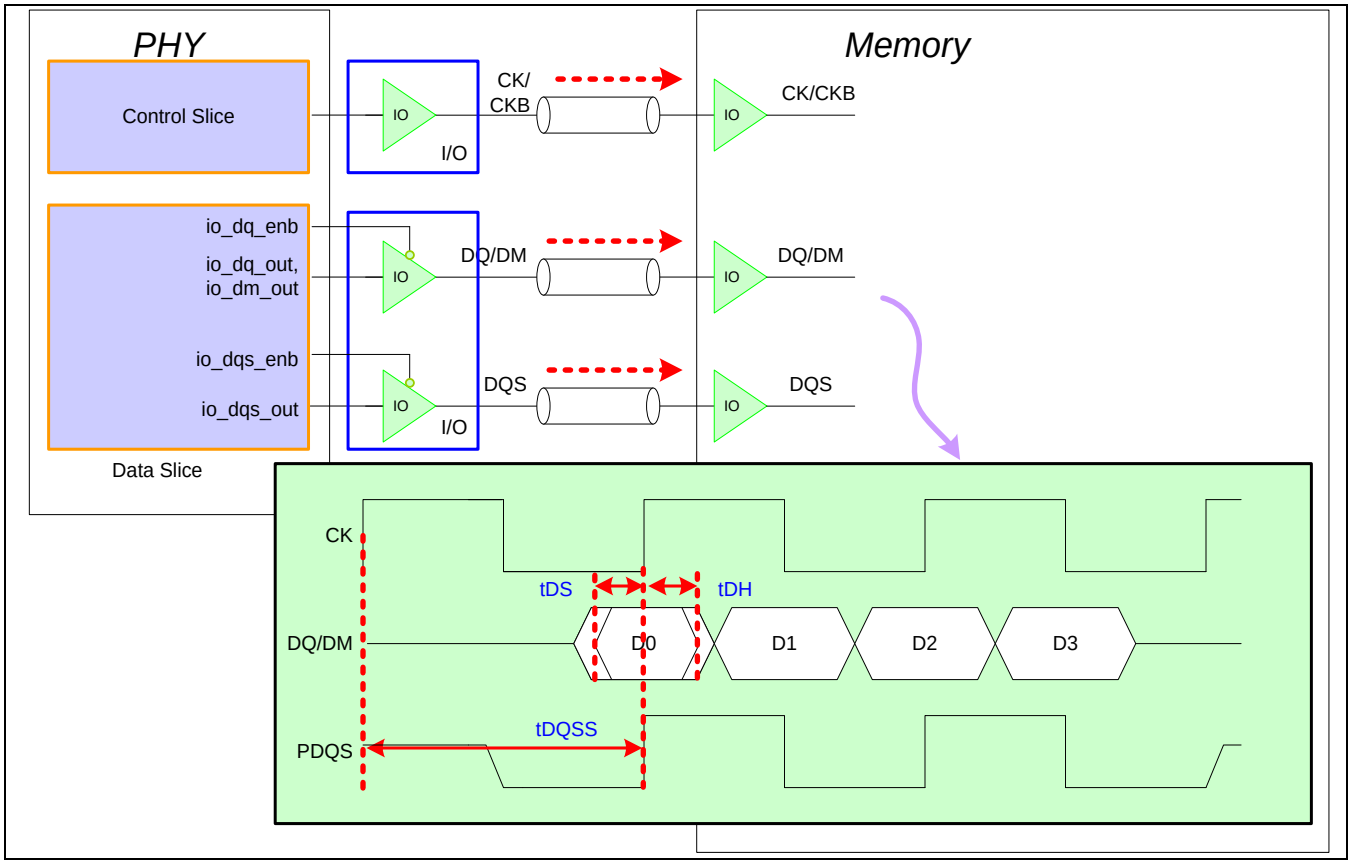

Figure 2-3. Block Diagram of the Write Path

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.

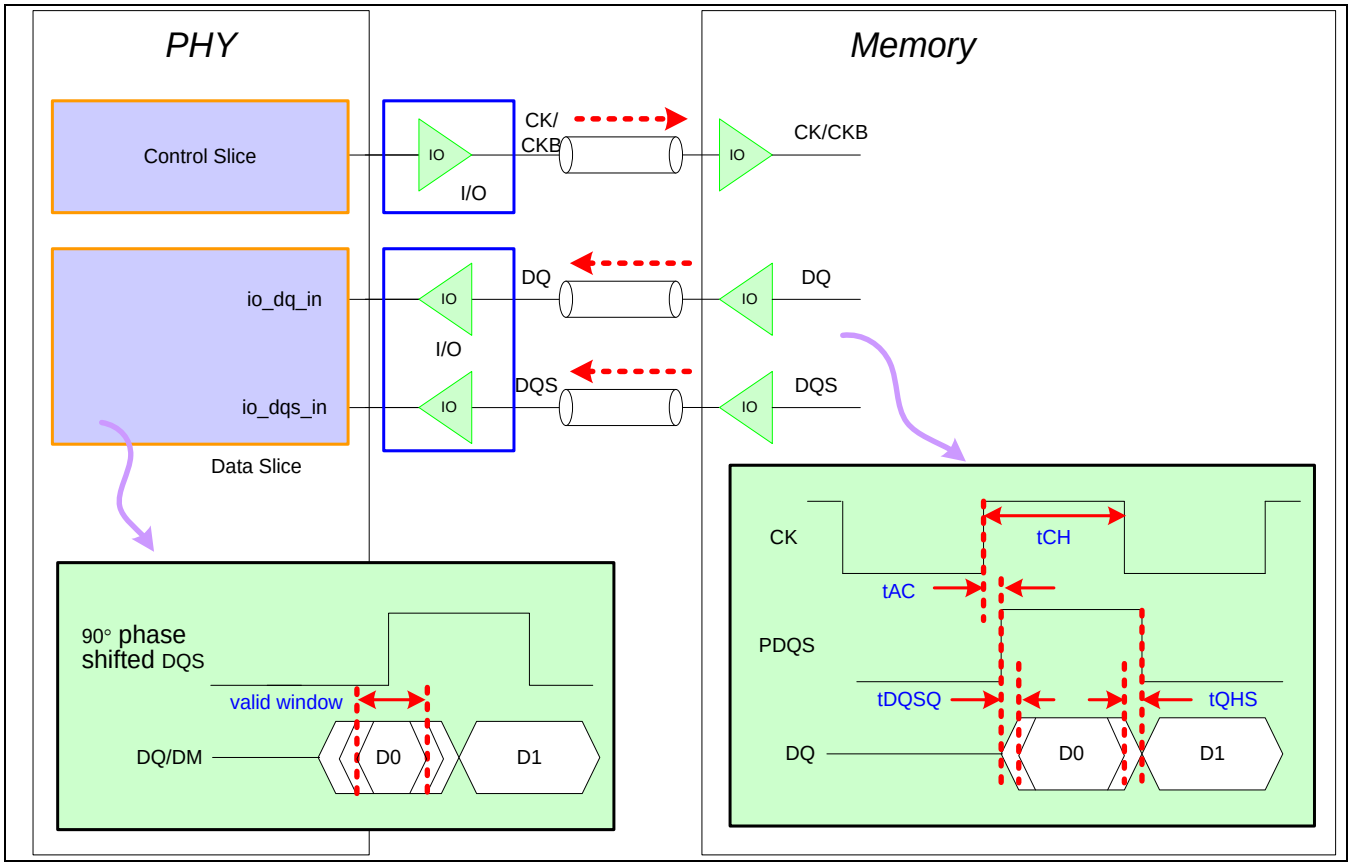

Figure 2-4. Block Diagram of the Read Path

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.

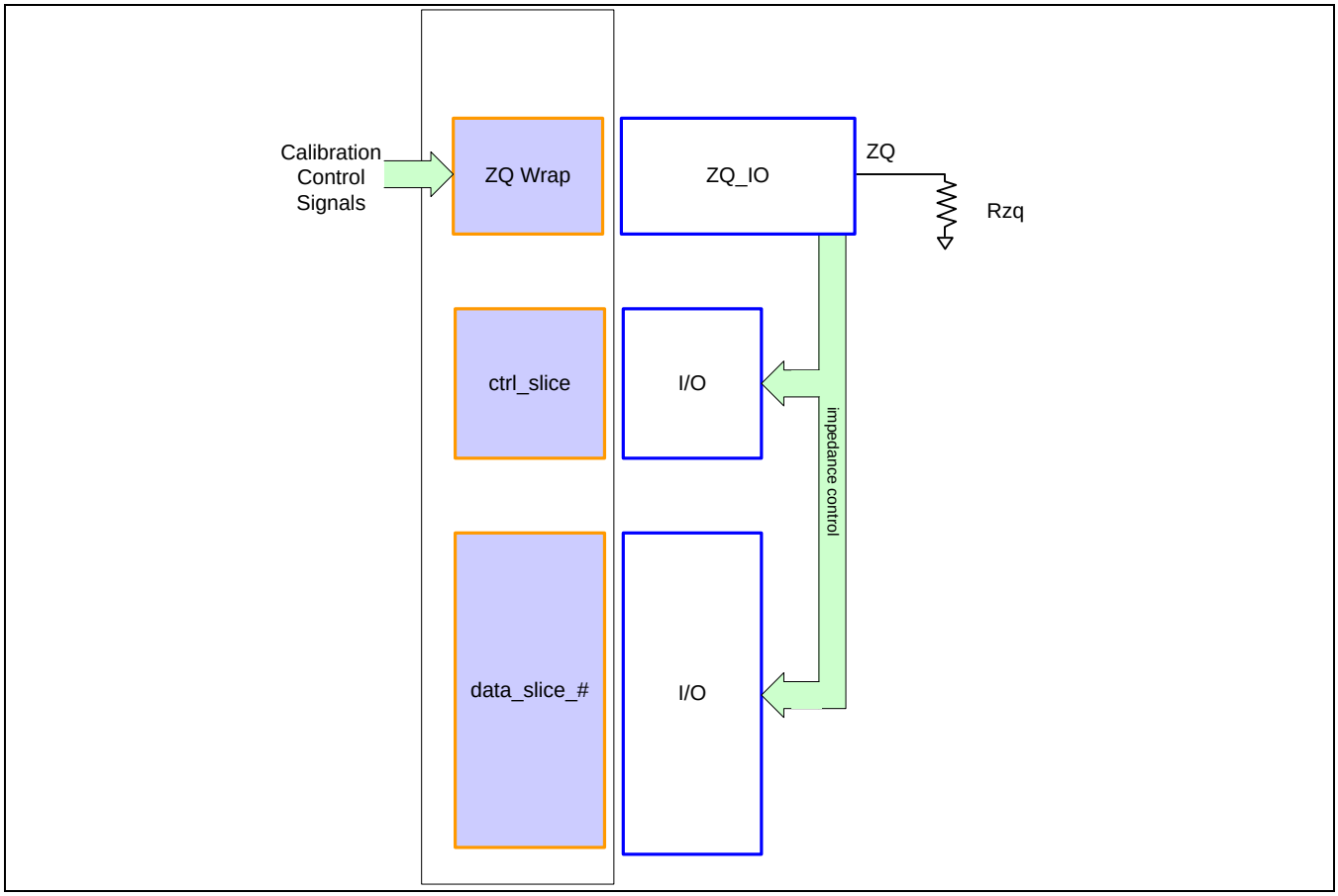

Figure 2-5. Connectivity Between ZQ I/O and the other I/Os

3 Signal Description

3.1 SYSTEM INTERFACE SIGNALS

Table 3-1. System Interface Signals

Name	Type	Description
clk2x	Input	0° phase system clock. It has strict jitter and duty requirements. Please refer to "UserGuide" for more details.
clk_en	Input	Clock Enable Signal to synchronize clk2x (PHY clock) with Controller Clock. 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 higher frequency than clk2x can be used for locking Master DLL for the low frequency operation. (p146) **NOTE:** All other input signals except for "clkm" should be generated in clk2x domain. The jitter and duty requirement of clkm is the same as that of clk2x.
rst_n	Input	LOW active reset signal. In normal operation mode, reset is latched by clk2x 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. 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[NS-1:0]	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[NS:0]	Input	This field controls toggling the input clock of 90 degree phase shift SDLL to save power. "ctrl_wakeup[NS]" 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[NS] is disabled). But if it doesn't use LPDDR2, "ctrl_wake_up[NS]" can be always deasserted. "ctrl_wakeup[NS-1:0]" for data_slice should be asserted 2 cycles prior to write command and then can be deasserted when ctrl_en falls. For example, if the number of "data_slice" is four, 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.
ctrl_place_type[NS:0]	Input	This field decides where each slice should be placed on chip. For example, if data_slice0 is placed on the top or bottom of chip, "ctrl_place_type[0]" should be zero and ioh_* type signals in DS0 should be used to connect with I/O.

NOTE: "NS" means the Number of "data_Slice"

3.2 APB INTERFACE SIGNALS

Table 3-2. Address & Command Interface Signals

Name	Type	Description
PCLK	Input	APB clock
PRESETn	Input	APB reset
PADDR[9:0]	Input	APB address
PSEL	Input	APB device select
PENABLE	Input	APB enable
PREADY	Output	APB ready
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[NS-1:0] dfi_wrdata_en_p1[NS-1:0]	Input	Write data and data mask enable.
dfi_wrdata_p0[2*8*NS-1:0] dfi_wrdata_p1[2*8*NS-1:0]	Input	Write data bus.
dfi_wrdata_mask_p0[2*NS-1:0] dfi_wrdata_mask_p0[2*NS-1:0]	Input	Write data byte mask.
dfi_rddata_en_p0[NS-1:0] dfi_rddata_en_p1[NS-1:0]	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[NS-1:0] dfi_rddata_valid_w1[NS-1:0]	Output	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[2*8*NS-1:0] dfi_rdata_w1[2*8*NS-1:0]	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[NS-1:0] ctrl_read_p1[NS-1:0]	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[NS-1:0], ctrl_gate_p1[NS-1:0]	Input	DQS clean signal.

NOTE: "NS" means the Number of "data_Slice"

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.

3.6 SSTL I/O INTERFACE SIGNALS

Table 3-6. SSTL I/O Horizontal Interface Signals

Name	Type	Description
ioh_adct_out[15:0]	Output	Address. These I/Os are used for ADDR
ioh_adct_en[15:0]	Output	Address I/O output enable
ioh_bank_out[2:0]	Output	Bank Address
ioh_bank_en[2:0]	Output	Bank Address I/O output enable
ioh_reset_out	Output	RESET for DDR3 Memory
ioh_reset_en	Output	RESET I/O output enable
ioh_odt_out[1:0]	Output	ODT
ioh_odt_en[1:0]	Output	ODT I/O output enable
ioh_cs_out[1:0]	Output	CS
ioh_cs_en[1:0]	Output	CS I/O output enable
ioh_ras_out	Output	RAS, row address selection.
ioh_ras_en	Output	RAS I/O output enable
ioh_cas_out	Output	CAS, column address selection.
ioh_cas_en	Output	CAS I/O output enable
ioh_wen_out	Output	WE, write enable.
ioh_wen_en	Output	WE I/O output enable
ioh_cke_out[1:0]	Output	CKE, clock enable.
ioh_cke_en[1:0]	Output	CKE I/O output enable
ioh_ck_out[1:0]	Output	CK/CKB differential clock
ioh_ck_en[1:0]	Output	CK I/O output enable
ioh_ckb_en[1:0]	Output	CKB I/O output enable
ioh_adct_pd[15:0]	Output	Address I/O Receiver Power Down Enable.
ioh_bank_pd[2:0]	Output	Bank Address I/O Receiver Power Down Enable.
ioh_reset_pd	Output	RESET I/O Receiver Power Down Enable.
ioh_odt_pd[1:0]	Output	ODT I/O Receiver Power Down Enable.
ioh_cs_pd[1:0]	Output	CS I/O Receiver Power Down Enable.
ioh_ras_pd	Output	RAS I/O Receiver Power Down Enable.
ioh_cas_pd	Output	CAS I/O Receiver Power Down Enable.
ioh_wen_pd	Output	WE I/O Receiver Power Down Enable.
ioh_ck_pd	Output	CK I/O Receiver Power Down Enable.
ioh_cke_pd[1:0]	Output	CKE I/O Receiver Power Down Enable.
ioh_adct_in[15:0]	Input	Address Input from I/O.
ioh_bank_in[2:0]	Input	Bank Address Input from I/O.
ioh_reset_in	Input	RESET Input from I/O.
ioh_odt_in[1:0]	Input	ODT Input from I/O.

Name	Type	Description
ioh_cs_in[1:0]	Input	CS Input from I/O.
ioh_ras_in	Input	RAS Input from I/O.
ioh_cas_in	Input	CAS Input from I/O.
ioh_wen_in	Input	WE Input from I/O.
ioh_ck_in	Input	CK Input from I/O.
ioh_cke_in[1:0]	Input	CKE Input from I/O.
ioh_dq_in[8*NS-1:0]	Input	DQ, Input Data.
ioh_dq_out[8*NS-1:0]	Output	DQ, Output Data.
ioh_dq_en[8*NS-1:0]	Output	DQ I/O output enable
ioh_dq_pdn[8*NS-1:0]	Output	DQ I/O Pull-down enable
ioh_dq_pup[8*NS-1:0]	Output	DQ I/O Pull-up enable
ioh_dq_read[8*NS-1:0]	Output	DQ I/O Drive tri-state enable
ioh_dq_cmosrcv[8*NS-1:0]	Output	DQ I/O CMOS/Differential receiver selection
ioh_dq_pd[8*NS-1:0]	Output	Power down pin for DQ I/O receiver
ioh_dm_out[NS-1:0]	Output	DM, Data mask.
ioh_dm_en[NS-1:0]	Output	DM I/O output enable
ioh_dm_in[NS-1:0]	Output	DM I/O Input Data
ioh_dm_pd[NS-1:0]	Output	Power down for DM I/O receiver
ioh_pdqs_out[NS-1:0]	Output	Positive DQS.
ioh_pdqs_en[NS-1:0]	Output	Positive DQS output enable
ioh_pdqs_pdn[NS-1:0]	Output	Positive DQS Pull-down enable
ioh_pdqs_pup[NS-1:0]	Output	Positive DQS Pull-up disable
ioh_pdqs_read[NS-1:0]	Output	Positive DQS Drive tri-state enable
ioh_ndqs_en[NS-1:0]	Output	Negative DQS output enable.
ioh_ndqs_pdn[NS-1:0]	Output	Negative DQS Pull-down enable
ioh_ndqs_pup[NS-1:0]	Output	Negative DQS Pull-up disable
ioh_ndqs_read[NS-1:0]	Output	Negative DQS Drive tri-state enable
ioh_adct_dds[47:0]	Output	Address Driver Strength Control
ioh_bank_dds[8:0]	Output	Bank Address Driver Strength Control
ioh_reset_dds[2:0]	Output	RESET Driver Strength Control
ioh_odt_dds[5:0]	Output	ODT Driver Strength Control
ioh_cs_dds[5:0]	Output	CS Driver Strength Control
ioh_ras_dds[2:0]	Output	RAS Driver Strength Control
ioh_cas_dds[2:0]	Output	CAS Driver Strength Control
ioh_wen_dds[2:0]	Output	WE Driver Strength Control
ioh_ck_dds[2:0]	Output	CK Driver Strength Control
ioh_cke_dds[5:0]	Output	CKE Driver Strength Control

Name	Type	Description
ioh_dq_dds[3*NS*8-1:0]	Output	DQ, Driver Strength Control
ioh_dm_dds[3*NS-1:0]	Output	DM Driver Strength Control
ioh_dqs_dds[3*NS-1:0]	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.

NOTE: "NS" means the Number of "data_Slice"

Table 3-7. SSTL I/O Vertical Interface Signals

Name	Type	Description
iov_adct_out[15:0]	Output	Address. These I/Os are used for ADDR
iov_adct_en[15:0]	Output	Address I/O output enable
iov_bank_out[2:0]	Output	Bank Address
iov_bank_en[2:0]	Output	Bank Address I/O output enable
iov_reset_out	Output	RESET for DDR3 Memory
iov_reset_en	Output	RESET I/O output enable
iov_odt_out[1:0]	Output	ODT
iov_odt_en[1:0]	Output	ODT I/O output enable
iov_cs_out[1:0]	Output	CS
iov_cs_en[1:0]	Output	CS I/O output enable
iov_ras_out	Output	RAS, row address selection.
iov_ras_en	Output	RAS I/O output enable

Name	Type	Description
iov_cas_out	Output	CAS, column address selection.
iov_cas_en	Output	CAS I/O output enable
iov_wen_out	Output	WE, write enable.
iov_wen_en	Output	WE I/O output enable
iov_cke_out[1:0]	Output	CKE, clock enable.
iov_cke_en[1:0]	Output	CKE I/O output enable
iov_ck_out[1:0]	Output	CK/CKB differential clock
iov_ck_en[1:0]	Output	CK I/O output enable
iov_ckb_en[1:0]	Output	CKB I/O output enable
iov_adct_pd[15:0]	Output	Address I/O Receiver Power Down Enable.
iov_bank_pd[2:0]	Output	Bank Address I/O Receiver Power Down Enable.
iov_reset_pd	Output	RESET I/O Receiver Power Down Enable.
iov_odt_pd[1:0]	Output	ODT I/O Receiver Power Down Enable.
iov_cs_pd[1:0]	Output	CS I/O Receiver Power Down Enable.
iov_ras_pd	Output	RAS I/O Receiver Power Down Enable.
iov_cas_pd	Output	CAS I/O Receiver Power Down Enable.
iov_wen_pd	Output	WE I/O Receiver Power Down Enable.
iov_ck_pd	Output	CK I/O Receiver Power Down Enable.
iov_cke_pd[1:0]	Output	CKE I/O Receiver Power Down Enable.
iov_adct_in[15:0]	Input	Address Input from I/O.
iov_bank_in[2:0]	Input	Bank Address Input from I/O.
iov_reset_in	Input	RESET Input from I/O.
iov_odt_in[1:0]	Input	ODT Input from I/O.
iov_cs_in[1:0]	Input	CS Input from I/O.
iov_ras_in	Input	RAS Input from I/O.
iov_cas_in	Input	CAS Input from I/O.
iov_wen_in	Input	WE Input from I/O.
iov_ck_in	Input	CK Input from I/O.
iov_cke_in[1:0]	Input	CKE Input from I/O.
iov_dq_in[8*NS-1:0]	Input	DQ, Input Data.
iov_dq_out[8*NS-1:0]	Output	DQ, Output Data.
iov_dq_en[8*NS-1:0]	Output	DQ I/O output enable
iov_dq_pdn[8*NS-1:0]	Output	DQ I/O Pull-down enable
iov_dq_pup[8*NS-1:0]	Output	DQ I/O Pull-up enable
iov_dq_read[8*NS-1:0]	Output	DQ I/O Drive tri-state enable
iov_dq_cmosrcv[8*NS-1:0]	Output	DQ I/O CMOS/Differential receiver selection
iov_dq_pd[8*NS-1:0]	Output	Power down pin for DQ I/O receiver

Name	Type	Description
iov_dm_out[NS-1:0]	Output	DM, Data mask.
iov_dm_en[NS-1:0]	Output	DM I/O output enable
iov_dm_in[NS-1:0]	Output	DM I/O Input Data
iov_dm_pd[NS-1:0]	Output	Power down for DM I/O receiver
iov_pdqs_out[NS-1:0]	Output	Positive DQS.
iov_pdqs_en[NS-1:0]	Output	Positive DQS output enable
iov_pdqs_pdn[NS-1:0]	Output	Positive DQS Pull-down enable
iov_pdqs_pup[NS-1:0]	Output	Positive DQS Pull-up disable
iov_pdqs_read[NS-1:0]	Output	Positive DQS Drive tri-state enable
iov_ndqs_en[NS-1:0]	Output	Negative DQS output enable.
iov_ndqs_pdn[NS-1:0]	Output	Negative DQS Pull-down enable
iov_ndqs_pup[NS-1:0]	Output	Negative DQS Pull-up disable
iov_ndqs_read[NS-1:0]	Output	Negative DQS Drive tri-state enable
iov_adct_dds[47:0]	Output	Address Driver Strength Control
iov_bank_dds[8:0]	Output	Bank Address Driver Strength Control
iov_reset_dds[2:0]	Output	RESET Driver Strength Control
iov_odt_dds[5:0]	Output	ODT Driver Strength Control
iov_cs_dds[5:0]	Output	CS Driver Strength Control
iov_ras_dds[2:0]	Output	RAS Driver Strength Control
iov_cas_dds[2:0]	Output	CAS Driver Strength Control
iov_wen_dds[2:0]	Output	WE Driver Strength Control
iov_ck_dds[2:0]	Output	CK Driver Strength Control
iov_cke_dds[5:0]	Output	CKE Driver Strength Control
iov_dq_dds[3*NS*8-1:0]	Output	DQ, Driver Strength Control
iov_dm_dds[3*NS-1:0]	Output	DM Driver Strength Control
iov_dqs_dds[3*NS-1:0]	Output	DQS Driver Strength Control

NOTE: "NS" means the Number of "data_Slice"

Table 3-8. ZQ I/O Interface Signals

Name	Type	Description
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

Name	Type	Description
		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.

3.7 DIRECT ACCESS INTERFACE SIGNALS

Table 3-9. 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[8*NS-1:0]	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[NS-1:0]	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[NS-1:0]	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[8*NS-1:0]	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[8*NS-1:0]	Output	When it's bypass/mux mode, ctrl_io_rdata is controlled by io*_dq_in.
ctrl_io_wdqs[NS-1:0]	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[NS-1:0]	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[NS-1:0]	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[NS-1:0]	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[NS-1:0]	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].
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.

Name	Type	Description
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.

NOTE: "*" means h or v. for example, io*_wen_en can mean ioh_wen_en or iov_wen_en.

NOTE: "NS" means the Number of "data_Slice"

3.8 FEEDBACK TEST INTERFACE SIGNALS

Table 3-10. 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_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[2:0]	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 DLL update. This value is limited by the half of the maximum delay in Master Delay Line. 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) 111: 3T/4(270° 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 :

Name	Type	Description
		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[NS:0]	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[NS:0]	Output	Feedback test stop with error for each slice.
test_oky[NS:0]	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_vwmc" test_ext_mode[1:0] = 2'b01 : The value of rdlvl_offsetw* will be on "test_ext_rdlvl_vwmc" 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),
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[8*NS-1:0]	Output	"Valid Window Margin"(VWM) after Read Leveling will be defined. It will be the center position in VWM.

NOTE: "NS" means the Number of "data_Slice"

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.

3.9 I/O TEST INTERFACE SIGNALS

Table 3-11. 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 DQ, PDQS, NDQS)
  - {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

3.10 SCAN TEST INTERFACE SIGNALS

Table 3-12. SCAN Test Interface Signals

Name	Type	Description
test_si[M-1:0]	Input	SCAN input signals for each scan-chain.(32-bit PHY) This field is 5-bit for 16-bit PHY.
test_so[M-1:0]	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.

NOTE: M means "scan chain nubmer". Please refer to User guide.

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 (p51)	0x1742_1E40
PHY_CON1	0x04	R/W	PHY Control Register 1 (p52)	0x2021_0100
PHY_CON2	0x08	R/W	PHY Control Register 2 (p53)	0x0001_0004
PHY_CON3	0x0C	R/W	PHY Control Register 3 (p54)	0x0000_0000
PHY_CON4	0x10	R/W	PHY Control Register 4 (p55)	0x0008_0000
PHY_CON5	0x14	R/W	PHY Control Register 5 (p55)	0x0000_0000
LP_CON0	0x18	R/W	Low Power Control Register 0 (p56)	0x0000_0000
RODT_CON0	0x1C	R/W	Read ODT Control Register 0 (p57)	0x0100_0000
OFFSETR_CON0	0x20	R/W	READ Code Control Register 0 (p57)	0x0808_0808
OFFSETR_CON1	0x24	R/W	READ Code Control Register 1 (p58)	0x0808_0808
OFFSETR_CON2	0x28	R/W	READ Code Control Register 2 (p59)	0x0000_0008
OFFSETW_CON0	0x30	R/W	WRITE Code Control Register 0 (p60)	0x0808_0808
OFFSETW_CON1	0x34	R/W	WRITE Code Control Register 1 (p61)	0x0808_0808
OFFSETW_CON2	0x38	R/W	WRITE Code Control Register 2 (p62)	0x0000_0008
OFFSETC_CON0	0x40	R/W	GATE Code Control Register 0 (p62)	0x0000_0000
OFFSETC_CON1	0x44	R/W	GATE Code Control Register 1 (p63)	0x0000_0000
OFFSETC_CON2	0x48	R/W	GATE Code Control Register 2 (p64)	0x0000_0000
SHIFTC_CON0	0x4C	R/W	GATE Code Shift Control Register 0 (p64)	0x0249_2492
OFFSETD_CON0	0x50	R/W	CMD Code Control Register (p66)	0x1000_0000
LP_DDR_CON0	0x58	R/W	Read Leveling Control Register 0 (p67)	0x0000_0208
LP_DDR_CON1	0x5C	R/W	Read Leveling Control Register 1 (p67)	0x0000_03FF
LP_DDR_CON2	0x60	R/W	Read Leveling Control Register 2 (p67)	0x0000_0000
LP_DDR_CON3	0x64	R/W	Read Leveling Control Register 3 (p68)	0x105E_107E
LP_DDR_CON4	0x68	R/W	Read Leveling Control Register 4 (p68)	0x0000_107F
WR_LVL_CON0	0x6C	R/W	Write Leveling Control Register 0 (p69)	0x0000_0000
WR_LVL_CON1	0x70	R/W	Write Leveling Control Register 1 (p69)	0x0000_0000
WR_LVL_CON2	0x74	R/W	Write Leveling Control Register 2 (p69)	0x0000_0000
WR_LVL_CON3	0x78	R/W	Write Leveling Control Register 3 (p69)	0x0000_0000
CA_DSKEW_CON0	0x7C	R/W	CA Deskew Control Register 0 (p69)	0x0000_0000

Register	Offset	R/W	Description	Initial Value
CA_DSKEW_CON1	0x80	R/W	CA Deskew Control Register 1 (p70)	0x0000_0000
CA_DSKEW_CON2	0x84	R/W	CA Deskew Control Register 2 (p70)	0x0000_0000
CA_DSKEW_CON3	0x88	R/W	CA Deskew Control Register 3 (p70)	0x0000_0000
CA_DSKEW_CON4	0x94	R/W	CA Deskew Control Register 4 (p70)	0x0000_0000
DRVDS_CON0	0x9C	R/W	Driver Strength Control Register 0 (p70)	0x0000_0000
DRVDS_CON1	0xA0	R/W	Driver Strength Control Register 1 (p71)	0x0000_0000
MDLL_CON0	0xB0	R/W	MDLL Control Register 0 (p71)	0x1010_0070
MDLL_CON1	0xB4	R	MDLL Control Register 1 (p72)	---
ZQ_CON0	0xC0	R/W	ZQ Control Register (p73)	0x0F00_7304
ZQ_CON1	0xC4	R	ZQ Status Register (p74)	---
ZQ_CON2	0xC8	R/W	ZQ Divider Control Register (p74)	0x0000_0007
ZQ_CON3	0xCC	R/W	ZQ Timer Control Register (p75)	0x0000_00F0
T_RDDATA_CON0	0xD0	R/W	Read Data Enable Status Register 0 (p75)	0x1555_5555
T_RDDATA_CON0	0xD4	R	Read Data Enable Status Register 1	---
T_RDDATA_CON0	0xD8	R	Read Data Enable Status Register 2	---
CAL_WL_STAT	0xDE	R	WL Calibration Fail Status Register	---
CAL_FAIL_STAT0	0xE0	R	Calibration Fail Status Register 0 (p76)	0x0000_0000
CAL_FAIL_STAT1	0xE4	R	Calibration Fail Status Register 1	0x0000_0000
CAL_FAIL_STAT2	0xE8	R	Calibration Fail Status Register 2	0x0000_0000
CAL_FAIL_STAT3	0xEC	R	Calibration Fail Status Register 3	0x0000_0000
CAL_GT_VWMC0	0xF0	R	Calibration Gate Training Centering Code 0	
CAL_GT_VWMC1	0xF4	R	Calibration Gate Training Centering Code 1	
CAL_GT_VWMC2	0xF8	R	Calibration Gate Training Centering Code 2	
CAL_GT_CYC	0xFC	R	Calibration Gate Training Cycle	
CAL_RD_VWMC0	0x100	R	Calibration Read Center Code 0	
CAL_RD_VWMC1	0x104	R	Calibration Read Center Code 1	
CAL_RD_VWMC2	0x108	R	Calibration Read Center Code 2	
CAL_RD_VWML0	0x110	R	Calibration Read Left Code 0	
CAL_RD_VWML1	0x114	R	Calibration Read Left Code 1	
CAL_RD_VWML2	0x118	R	Calibration Read Left Code 2	
CAL_RD_VWMR0	0x120	R	Calibration Read Right Code 0	
CAL_RD_VWMR1	0x124	R	Calibration Read Right Code 1	
CAL_RD_VWMR2	0x128	R	Calibration Read Right Code 2	
CAL_WR_VWMC0	0x130	R	Calibration Write Center Code 0	
CAL_WR_VWMC1	0x134	R	Calibration Write Center Code 1	
CAL_WR_VWMC2	0x138	R	Calibration Write Center Code 2	
CAL_WR_VWML0	0x140	R	Calibration Write Left Code 0	

Register	Offset	R/W	Description	Initial Value
CAL_WR_VWML1	0x144	R	Calibration Write Left Code 1	
CAL_WR_VWML2	0x148	R	Calibration Write Left Code 2	
CAL_WR_VWMR0	0x150	R	Calibration Write Right Code 0	
CAL_WR_VWMR1	0x154	R	Calibration Write Right Code 1	
CAL_WR_VWMR2	0x158	R	Calibration Write Right Code 2	
CAL_DM_VWMC0	0x160	R	Calibration DM Center Code 0	
CAL_DM_VWMC1	0x164	R	Calibration DM Center Code 1	
CAL_DM_VWMC2	0x168	R	Calibration DM Center Code 2	
CAL_DM_VWML0	0x170	R	Calibration DM Left Code 0	
CAL_DM_VWML1	0x174	R	Calibration DM Left Code 1	
CAL_DM_VWML2	0x178	R	Calibration DM Left Code 2	
CAL_DM_VWMR0	0x180	R	Calibration DM Right Code 0	
CAL_DM_VWMR1	0x184	R	Calibration DM Right Code 1	
CAL_DM_VWMR2	0x188	R	Calibration DM Right Code 2	
RD_DESKEW_CON0	0x190	R/W	READ DE-SKEW CONTROL 0	
RD_DESKEW_CON1	0x194	R/W	READ DE-SKEW CONTROL 1	
RD_DESKEW_CON2	0x198	R/W	READ DE-SKEW CONTROL 2	
RD_DESKEW_CON3	0x19C	R/W	READ DE-SKEW CONTROL 3	
RD_DESKEW_CON4	0x1A0	R/W	READ DE-SKEW CONTROL 4	
RD_DESKEW_CON5	0x1A4	R/W	READ DE-SKEW CONTROL 5	
RD_DESKEW_CON6	0x1A8	R/W	READ DE-SKEW CONTROL 6	
RD_DESKEW_CON7	0x1AC	R/W	READ DE-SKEW CONTROL 7	
RD_DESKEW_CON8	0x1B0	R/W	READ DE-SKEW CONTROL 8	
RD_DESKEW_CON9	0x1B4	R/W	READ DE-SKEW CONTROL 9	
RD_DESKEW_CON10	0x1B8	R/W	READ DE-SKEW CONTROL 10	
RD_DESKEW_CON11	0x1BC	R/W	READ DE-SKEW CONTROL 11	
RD_DESKEW_CON12	0x1C0	R/W	READ DE-SKEW CONTROL 12	
RD_DESKEW_CON13	0x1C4	R/W	READ DE-SKEW CONTROL 13	
RD_DESKEW_CON14	0x1C8	R/W	READ DE-SKEW CONTROL 14	
RD_DESKEW_CON15	0x1CC	R/W	READ DE-SKEW CONTROL 15	
RD_DESKEW_CON16	0x1D0	R/W	READ DE-SKEW CONTROL 16	
RD_DESKEW_CON17	0x1D4	R/W	READ DE-SKEW CONTROL 17	
RD_DESKEW_CON18	0x1D8	R/W	READ DE-SKEW CONTROL 18	
RD_DESKEW_CON19	0x1DC	R/W	READ DE-SKEW CONTROL 19	
RD_DESKEW_CON20	0x1E0	R/W	READ DE-SKEW CONTROL 20	
RD_DESKEW_CON21	0x1E4	R/W	READ DE-SKEW CONTROL 21	
RD_DESKEW_CON22	0x1E8	R/W	READ DE-SKEW CONTROL 22	

Register	Offset	R/W	Description	Initial Value
RD_DESKEW_CON23	0x1EC	R/W	READ DE-SKEW CONTROL 23	
WR_DESKEW_CON0	0x1F0	R/W	WRITE DE-SKEW CONTROL 0	
WR_DESKEW_CON1	0x1F4	R/W	WRITE DE-SKEW CONTROL 1	
WR_DESKEW_CON2	0x1F8	R/W	WRITE DE-SKEW CONTROL 2	
WR_DESKEW_CON3	0x1FC	R/W	WRITE DE-SKEW CONTROL 3	
WR_DESKEW_CON4	0x200	R/W	WRITE DE-SKEW CONTROL 4	
WR_DESKEW_CON5	0x204	R/W	WRITE DE-SKEW CONTROL 5	
WR_DESKEW_CON6	0x208	R/W	WRITE DE-SKEW CONTROL 6	
WR_DESKEW_CON7	0x20C	R/W	WRITE DE-SKEW CONTROL 7	
WR_DESKEW_CON8	0x210	R/W	WRITE DE-SKEW CONTROL 8	
WR_DESKEW_CON9	0x214	R/W	WRITE DE-SKEW CONTROL 9	
WR_DESKEW_CON10	0x218	R/W	WRITE DE-SKEW CONTROL 10	
WR_DESKEW_CON11	0x21C	R/W	WRITE DE-SKEW CONTROL 11	
WR_DESKEW_CON12	0x220	R/W	WRITE DE-SKEW CONTROL 12	
WR_DESKEW_CON13	0x224	R/W	WRITE DE-SKEW CONTROL 13	
WR_DESKEW_CON14	0x228	R/W	WRITE DE-SKEW CONTROL 14	
WR_DESKEW_CON15	0x22C	R/W	WRITE DE-SKEW CONTROL 15	
WR_DESKEW_CON16	0x230	R/W	WRITE DE-SKEW CONTROL 16	
WR_DESKEW_CON17	0x234	R/W	WRITE DE-SKEW CONTROL 17	
WR_DESKEW_CON18	0x238	R/W	WRITE DE-SKEW CONTROL 18	
WR_DESKEW_CON19	0x23C	R/W	WRITE DE-SKEW CONTROL 19	
WR_DESKEW_CON20	0x240	R/W	WRITE DE-SKEW CONTROL 20	
WR_DESKEW_CON21	0x244	R/W	WRITE DE-SKEW CONTROL 21	
WR_DESKEW_CON22	0x248	R/W	WRITE DE-SKEW CONTROL 22	
WR_DESKEW_CON23	0x24C	R/W	WRITE DE-SKEW CONTROL 23	
DM_DESKEW_CON0	0x250	R/W	DM DE-SKEW CONTROL 0	
DM_DESKEW_CON1	0x254	R/W	DM DE-SKEW CONTROL 1	
DM_DESKEW_CON2	0x258	R/W	DM DE-SKEW CONTROL 2	
VWMC_STAT0	0x25C	R	VWMC STAT 0	
VWMC_STAT1	0x260	R	VWMC STAT 1	
VWMC_STAT2	0x264	R	VWMC STAT 2	
VWMC_STAT3	0x268	R	VWMC STAT 3	
VWMC_STAT4	0x26C	R	VWMC STAT 4	
VWMC_STAT5	0x270	R	VWMC STAT 5	
VWMC_STAT6	0x274	R	VWMC STAT 6	
VWMC_STAT7	0x278	R	VWMC STAT 7	
VWMC_STAT8	0x27C	R	VWMC STAT 8	

Register	Offset	R/W	Description	Initial Value
VWMC_STAT9	0x280	R	VWMC STAT 9	
VWMC_STAT10	0x284	R	VWMC STAT 10	
VWMC_STAT11	0x288	R	VWMC STAT 11	
VWMC_STAT12	0x28C	R	VWMC STAT 12	
VWMC_STAT13	0x290	R	VWMC STAT 13	
VWMC_STAT14	0x294	R	VWMC STAT 14	
VWMC_STAT15	0x298	R	VWMC STAT 15	
VWMC_STAT16	0x29C	R	VWMC STAT 16	
VWMC_STAT17	0x2A0	R	VWMC STAT 17	
VWMC_STAT18	0x2A4	R	VWMC STAT 18	
VWMC_STAT19	0x2A8	R	VWMC STAT 19	
VWMC_STAT20	0x2AC	R	VWMC STAT 20	
VWMC_STAT21	0x2B0	R	VWMC STAT 21	
VWMC_STAT22	0x2B4	R	VWMC STAT 22	
VWMC_STAT23	0x2B8	R	VWMC STAT 23	
DM_VWMC_STAT0	0x2BC	R	DM VWMC 0	
DM_VWMC_STAT1	0x2C0	R	DM VWMC 1	
DM_VWMC_STAT2	0x2C4	R	DM VWMC 2	
VWML_STAT0	0x2C8	R	VWML STAT 0	
VWML_STAT1	0x2CC	R	VWML STAT 1	
VWML_STAT2	0x2D0	R	VWML STAT 2	
VWML_STAT3	0x2D4	R	VWML STAT 3	
VWML_STAT4	0x2D8	R	VWML STAT 4	
VWML_STAT5	0x2DC	R	VWML STAT 5	
VWML_STAT6	0x2E0	R	VWML STAT 6	
VWML_STAT7	0x2E4	R	VWML STAT 7	
VWML_STAT8	0x2E8	R	VWML STAT 8	
VWML_STAT9	0x2EC	R	VWML STAT 9	
VWML_STAT10	0x2F0	R	VWML STAT 10	
VWML_STAT11	0x2F4	R	VWML STAT 11	
VWML_STAT12	0x2F8	R	VWML STAT 12	
VWML_STAT13	0x2FC	R	VWML STAT 13	
VWML_STAT14	0x300	R	VWML STAT 14	
VWML_STAT15	0x304	R	VWML STAT 15	
VWML_STAT16	0x308	R	VWML STAT 16	
VWML_STAT17	0x30C	R	VWML STAT 17	
VWML_STAT18	0x310	R	VWML STAT 18	

Register	Offset	R/W	Description	Initial Value
VWML_STAT19	0x314	R	VWML STAT 19	
VWML_STAT20	0x318	R	VWML STAT 20	
VWML_STAT21	0x31C	R	VWML STAT 21	
VWML_STAT22	0x320	R	VWML STAT 22	
VWML_STAT23	0x324	R	VWML STAT 23	
DM_VWML_STAT0	0x328	R	DM VWML 0	
DM_VWML_STAT1	0x32C	R	DM VWML 1	
DM_VWML_STAT2	0x330	R	DM VWML 2	
VWMR_STAT0	0x334	R	VWMR STAT 0	
VWMR_STAT1	0x338	R	VWMR STAT 1	
VWMR_STAT2	0x33C	R	VWMR STAT 2	
VWMR_STAT3	0x340	R	VWMR STAT 3	
VWMR_STAT4	0x344	R	VWMR STAT 4	
VWMR_STAT5	0x348	R	VWMR STAT 5	
VWMR_STAT6	0x34C	R	VWMR STAT 6	
VWMR_STAT7	0x350	R	VWMR STAT 7	
VWMR_STAT8	0x354	R	VWMR STAT 8	
VWMR_STAT9	0x358	R	VWMR STAT 9	
VWMR_STAT10	0x35C	R	VWMR STAT 10	
VWMR_STAT11	0x360	R	VWMR STAT 11	
VWMR_STAT12	0x364	R	VWMR STAT 12	
VWMR_STAT13	0x368	R	VWMR STAT 13	
VWMR_STAT14	0x36C	R	VWMR STAT 14	
VWMR_STAT15	0x370	R	VWMR STAT 15	
VWMR_STAT16	0x374	R	VWMR STAT 16	
VWMR_STAT17	0x378	R	VWMR STAT 17	
VWMR_STAT18	0x37C	R	VWMR STAT 18	
VWMR_STAT19	0x380	R	VWMR STAT 19	
VWMR_STAT20	0x384	R	VWMR STAT 20	
VWMR_STAT21	0x388	R	VWMR STAT 21	
VWMR_STAT22	0x38C	R	VWMR STAT 22	
VWMR_STAT23	0x390	R	VWMR STAT 23	
DM_VWMR_STAT0	0x394	R	DM VWMR 0	
DM_VWMR_STAT1	0x398	R	DM VWMR 1	
DM_VWMR_STAT2	0x39C	R	DM VWMR 2	
DQ_IO_RDATA0	0x3A0	R	DQ I/O Read Data STAT 0	
DQ_IO_RDATA1	0x3A4	R	DQ I/O Read Data STAT 1	

Register	Offset	R/W	Description	Initial Value
DQ_IO_RDATA2	0x3A8	R	DQ I/O Read Data STAT 2	
VERSION_INFO	0x3AC	R	Version Information(p109)	

4.2 PHY Control Register

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

Field	Bit	R/W	Description	Initial Value
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_mode	[23:22]	R/W	It controls when DLL is updated. 2'b00 : Update always 2'b01 : To update depending on "ctrl_flock" 2'b10 : To update depending on "ctrl_clock" 2'b01 : Don't update DLL	2'b01
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 : Update when difference is greater than 0 2'b01 : Update when difference is greater than 3 2'b10 : Update when difference is greater than 7 2'b11 : Update when difference is greater than 15	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'b001) 3'b110 : tWTR = 4 cycles	3'b001
wrlvl_mode	[16]	R/W	Write Leveling Mode 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 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
ctrl_wr_dis	[10]	R/W	Write ODT(On-Die-Termination) Disable Signal during Write Calibration. 1'b0 : drive io*_odt_out(=ODT) to 1 during Write Calibration. 1'b1 : drive io*_odt_out(=ODT) to 0 during Write Calibration.	1'b1
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	1'b0

Field	Bit	R/W	Description	Initial Value
			mode, ctrl_pulld_dqs[3:0] == 4'b1111, N = 0,1,2…) 1'b1: gate signal length = "burst length / 2" – 1	
~~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_dis~~	~~[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'b111: Power Down Mode for SSTL I/O	3'b000

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'h2
~~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'h1~~
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. 16'hFF00 : DDR3 16'h00FF : LPDDR3	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_cal_start	[27]	R/W	DQ Calibration Start Signal to align DQ, DM during write.	1'b0
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
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
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
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'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"	7'b01
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	1'b0

Field	Bit	R/W	Description	Initial Value
			whenever "dfi_ctrlupd_req" or "dfi_phyupd_req" is issued during DLLDeskewEn=1. It is required to compensate On-chip VT variation.	
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.	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 exapmle, 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_CON3(Address Offset=0x0C)

Field	Bit	R/W	Description	Initial Value
wl_cal_resp	[27]	R/W	Response after Write Leveling Calibration	0x0
rd_wr_cal_resp	[26]	R/W	Response after Read or Write Calibration	0x0
wrlvl_resp	[24]	R/W	Response after Write Leveling	0x0
wl_cal_start	[21]	R/W	Start Write Leveling Calibration	0x0
wl_cal_mode	[20]	R/W	Write Leveling Calibration mode Enable	0x0
rd_cal_start	[19]	R/W	Start Read Calibration signal. It should be disabled after rd_wr_cal_resp is enabled.	0x0
gate_lvl_start	[18]	R/W	Start Gate Leveling signal. It should be disabled after Gate Leveling Response is asserted.	0x0

Field	Bit	R/W	Description	Initial Value
wrlvl_start	[16]	R/W	Start Write Leveling signal. It can be enabled when wrlvl_mode = 1. It should be disabled after Write Leveling Response is asserted.	0x0
reg_mode	[7:0]	R/W	Register mode control to write the information at each data slice. Please refer to RD_DESKEW_CON* or WR_DESKEW_CON* register(p87, p93)	0x0

Table 4-6. PHY_CON4(Address Offset=0x10)

Field	Bit	R/W	Description	Initial Value
ctrl_wrlat	[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.	5'h8
ctrl_bstlen	[12:8]	R/W	Burst Length(BL)	5'h0
ctrl_rdlat	[4:0]	R/W	Read Latency(RL)	5'h0

Table 4-7. PHY_CON5(Address Offset=0x14)

Field	Bit	R/W	Description	Initial Value
ctrl_wrlat_plus9	[27]	R/W	This field can control Write Latency(WL) by one cycle. [0] : Write Latency Decreases by half clock cycle when enabled.	1'h0
ctrl_wrlat_plus8	[26:24]	R/W	This field can control Write Latency(WL) by half cycle, one or two cycles for Data_Slice8. [0] : Write Latency Increases by half clock cycle when enabled. [1] : Write Latency Increases by one clock cycle when enabled. [2] : Write Latency Increases by two clock cycle when enabled.	3'h0
ctrl_wrlat_plus7	[23:21]	R/W	This field can control Write Latency(WL) by half cycle, one or two cycles for Data_Slice7. [0] : Write Latency Increases by half clock cycle when enabled. [1] : Write Latency Increases by one clock cycle when enabled. [2] : Write Latency Increases by two clock cycle when enabled.	3'h0
ctrl_wrlat_plus6	[20:18]	R/W	This field can control Write Latency(WL) by half cycle, one or two cycles for Data_Slice6. [0] : Write Latency Increases by half clock cycle when enabled. [1] : Write Latency Increases by one clock cycle when enabled. [2] : Write Latency Increases by two clock cycle when enabled.	3'h0
ctrl_wrlat_plus5	[17:15]	R/W	This field can control Write Latency(WL) by half cycle, one or two cycles for Data_Slice5. [0] : Write Latency Increases by half clock cycle when enabled. [1] : Write Latency Increases by one clock cycle when enabled. [2] : Write Latency Increases by two clock cycle when enabled.	3'h0

Field	Bit	R/W	Description	Initial Value
ctrl_wrlat_plus4	[14:12]	R/W	This field can control Write Latency(WL) by half cycle, one or two cycles for Data_Slice4. [0] : Write Latency Increases by half clock cycle when enabled. [1] : Write Latency Increases by one clock cycle when enabled. [2] : Write Latency Increases by two clock cycle when enabled.	3'h0
ctrl_wrlat_plus3	[11:9]	R/W	This field can control Write Latency(WL) by half cycle, one or two cycles for Data_Slice3. [0] : Write Latency Increases by half clock cycle when enabled. [1] : Write Latency Increases by one clock cycle when enabled. [2] : Write Latency Increases by two clock cycle when enabled.	3'h0
ctrl_wrlat_plus2	[8:6]	R/W	This field can control Write Latency(WL) by half cycle, one or two cycles for Data_Slice2. [0] : Write Latency Increases by half clock cycle when enabled. [1] : Write Latency Increases by one clock cycle when enabled. [2] : Write Latency Increases by two clock cycle when enabled.	3'h0
ctrl_wrlat_plus1	[5:3]	R/W	This field can control Write Latency(WL) by half cycle, one or two cycles for Data_Slice1. [0] : Write Latency Increases by half clock cycle when enabled. [1] : Write Latency Increases by one clock cycle when enabled. [2] : Write Latency Increases by two clock cycle when enabled.	3'h0
ctrl_wrlat_plus0	[2:0]	R/W	This field can control Write Latency(WL) by half cycle, one or two cycles for Data_Slice0. [0] : Write Latency Increases by half clock cycle when enabled. [1] : Write Latency Increases by one clock cycle when enabled. [2] : Write Latency Increases by two clock cycle when enabled.	3'h0

NOTE: When "wl_cal_mode"=1'b1, "PHY_CON5" will show the results of HW Write Latency Calibration. Please don't change "PHY_CON5" during "wl_cal_mode"=1'b1.

4.3 Low Power Control Register

Table 4-8. LP_CON0(Address Offset=0x18)

Field	Bit	R/W	Description	Initial Value
ctrl_pulld_dq	[24:16]	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	[8:0]	R/W	Active HIGH signal to pull-up or down PDQS/NDQS signals. 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'h0

4.4 Read ODT Control Register

Table 4-9. RODT_CON0(Address Offset=0x1C)

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
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'h1
ctrl_read_dis	~~[5]~~ [16]	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_read_width	[NS-1:0]	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

NOTE: NS means the number of slice.

4.5 Offset Read Control Register

Caution: Be careful that "ctrl_offsetr*" can be used for the other purpose(=Read Deskew Code Register). Please refer to read_mode_con(=PHY_CON3[7:0]).

Table 4-10. OFFSETR_CON0(Address Offset=0x20)

Field	Bit	R/W	Description	Initial Value
ctrl_offsetr3	[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	[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

Table 4-11. OFFSETR_CON1(Address Offset=0x24)

Field	Bit	R/W	Description	Initial Value
ctrl_offsetr7	[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_offsetr7[7] = 1: (tFS: fine step delay) Read DQS 90° delay amount – ctrl_offsetr7[6:0] x tFS ctrl_offsetr7[7] = 0: Read DQS 90° delay amount + ctrl_offsetr7[6:0] x tFS	0x8

Field	Bit	R/W	Description	Initial Value
ctrl_offsetr6	[23:16]	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_offsetr6[7] = 1: (tFS: fine step delay) Read DQS 90° delay amount – ctrl_offsetr6[6:0] x tFS ctrl_offsetr6[7] = 0: Read DQS 90° delay amount + ctrl_offsetr6[6:0] x tFS	0x8
ctrl_offsetr5	[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_offsetr5[7] = 1: (tFS: fine step delay) Read DQS 90° delay amount – ctrl_offsetr5[6:0] x tFS ctrl_offsetr5[7] = 0: Read DQS 90° delay amount + ctrl_offsetr5[6:0] x tFS	0x8
ctrl_offsetr4	[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_offsetr4[7] = 1: (tFS: fine step delay) Read DQS 90° delay amount – ctrl_offsetr4[6:0] x tFS ctrl_offsetr4[7] = 0: Read DQS 90° delay amount + ctrl_offsetr4[6:0] x tFS	0x8

Table 4-12. OFFSETR_CON2(Address offset=0x28)

Field	Bit	R/W	Description	Initial Value
ctrl_offsetr8	[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_offsetr8[7] = 1: (tFS: fine step delay) Read DQS 90° delay amount – ctrl_offsetr8[6:0] x tFS ctrl_offsetr8[7] = 0:	0x8

Field	Bit	R/W	Description	Initial Value
			Read DQS 90° delay amount + ctrl_offsetr8[6:0] x tFS	

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.

4.6 Offset Write Code Control Register

Caution: Be careful that "ctrl_offsetw*" can be used for the other purpose (=Write Deskew Code Register). Please refer to reg_mode (=PHY_CON3[7:0]).

Table 4-13. OFFSETW_CON0(Address Offset=0x30)

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.	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. 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	
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) 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

Table 4-14. OFFSETW_CON1(Address Offset=0x34)

Field	Bit	R/W	Description	Initial Value
ctrl_offsetw7	[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_offsetw7[7] = 1 : (tFS: fine step delay) Write DQ 270° delay amount – ctrl_offsetw7[6:0] x tFS ctrl_offsetw7[7] = 0 : Write DQ 270° delay amount + ctrl_offsetw7[6:0] x tFS	0x8
ctrl_offsetw6	[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_offsetw6[7] = 1 : (tFS: fine step delay) Write DQ 270° delay amount – ctrl_offsetw6[6:0] x tFS ctrl_offsetw6[7] = 0 : Write DQ 270° delay amount + ctrl_offsetw6[6:0] x tFS	0x8

Field	Bit	R/W	Description	Initial Value
ctrl_offsetw5	[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_offsetw5[7] = 1 : (tFS: fine step delay) Write DQ 270° delay amount – ctrl_offsetw5[6:0] x tFS ctrl_offsetw5[7] = 0 : Write DQ 270° delay amount + ctrl_offsetw5[6:0] x tFS	0x8
ctrl_offsetw4	[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_offsetw4[7] = 1 : (tFS: fine step delay) Write DQ 270° delay amount – ctrl_offsetw4[6:0] x tFS ctrl_offsetw4[7] = 0 : Write DQ 270° delay amount + ctrl_offsetw4[6:0] x tFS	0x8

Table 4-15. OFFSETW_CON2(Address offset=0x38)

Field	Bit	R/W	Description	Initial Value
ctrl_offsetw8	[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_offsetw8[7] = 1 : (tFS: fine step delay) Write DQ 270° delay amount – ctrl_offsetw8[6:0] x tFS ctrl_offsetw8[7] = 0 : Write DQ 270° delay amount + ctrl_offsetw8[6:0] x tFS	0x8

4.7 GATE Code Control Register

Table 4-16. OFFSETC_CON0(Address Offset=0x40)

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	0x0

Field	Bit	R/W	Description	Initial Value
			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	
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. 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_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 4-17. OFFSETC_CON1(Address Offset=0x44)

Field	Bit	R/W	Description	Initial Value
ctrl_offsetc7	[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_offsetc6	[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	0x0

Field	Bit	R/W	Description	Initial Value
			ctrl_offsetc [7] = 0 : GATEout delay amount + ctrl_offsetc [6:0] x tFS	
ctrl_offsetc5	[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. 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_offsetc4	[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 4-18. OFFSETC_CON2(Address Offset=0x48)

Field	Bit	R/W	Description	Initial Value
ctrl_offsetc8	[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 4-19. SHIFTC_CON0(Address Offset=0x4C)

Field	Bit	R/W	Description	Initial Value
ctrl_shiftc8	[26:24]	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_shiftc7	[23:21]	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_shiftc6	[20:18]	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_shiftc5	[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_shiftc4	[14:12]	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	[11:9]	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	[8:6]	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	3'b010

Field	Bit	R/W	Description	Initial Value
			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	[5:3]	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

4.8 Command SDLL Code Control Register

Table 4-20. OFFSETD_CON0(Address Offset=0x50)

Field	Bit	R/W	Description	Initial Value
upd_mode	[28]	R/W	This field controls "PHY Update" Mode. (Refer to 8.7) 1'b1 : MC-Initiated Update Mode 1'b0 : PHY-Initiated Update Mode	0x1
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 "DLL Code Update"(p156) to use ctrl_resync.	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: ctrl_offsetd[7] = 1 : (tFS: fine step delay) 270° delay amount – ctrl_offsetd[6:0] x tFS ctrl_offsetd[7] = 0 :	0x8

Field	Bit	R/W	Description	Initial Value
			270° delay amount + ctrl_offsetd[6:0] x tFS	

4.9 Calibration Control Register

Table 4-21. LP_DDR_CON0(Address Offset=0x58)

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 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)	0x208

Table 4-22. LP_DDR_CON1(Address Offset=0x5C)

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

Table 4-23. LP_DDR_CON2(Address Offset=0x60)

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]	0x0

Field	Bit	R/W	Description	Initial Value
			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.	

Table 4-24. LP_DDR_CON3(Address Offset=0x64)

Field	Bit	R/W	Description	Initial Value
~~ddr3_cmd~~	~~[29:16]~~	~~R/W~~	~~DDR3 Command~~	~~0x105E~~
cmd_active	[12:0]	R/W	This field means "Read command" which should be executed during calibration. "16'h000E" : LPDDR2 or LPDDR3. "16'h105E" : DDR3 [1:0] : CS[1:0] [2:3] : CKE[1:0] [4] : WEN [5] : CAS [6] : RAS(should be always "1") [8:7] : ODT[1:0] (Not applicable) [11:9] : BANK[2:0] [12] : RESET	0x105E

Table 4-25. LP_DDR_CON4(Address Offset=0x68)

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	[12:0]	R/W	Default Command 16'h000F(LPDDR2, LPDDR3) 16'h107F(DDR2, DDR3) [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	0x107F

4.10 Write Leveling Control Register

Table 4-26. WR_LVL_CON0(Address Offset=0x6C)

Field	Bit	R/W	Description	Initial Value
ctrl_wrlvl3_code	[31:24]	R/W	Write Level Slave DLL Code Value for Data_Slice 3	0x0
ctrl_wrlvl2_code	[23:16]	R/W	Write Level Slave DLL Code Value for Data_Slice 2	0x0
ctrl_wrlvl1_code	[15:8]	R/W	Write Level Slave DLL Code Value for Data_Slice 1	0x0
ctrl_wrlvl0_code	[7:0]	R/W	Write Level Slave DLL Code Value for Data_Slice 0	0x0

Table 4-27. WR_LVL_CON1(Address Offset=0x70)

Field	Bit	R/W	Description	Initial Value
ctrl_wrlvl7_code	[31:24]	R/W	Write Level Slave DLL Code Value for Data_Slice 7	0x0
ctrl_wrlvl6_code	[23:16]	R/W	Write Level Slave DLL Code Value for Data_Slice 6	0x0
ctrl_wrlvl5_code	[15:8]	R/W	Write Level Slave DLL Code Value for Data_Slice 5	0x0
ctrl_wrlvl4_code	[7:0]	R/W	Write Level Slave DLL Code Value for Data_Slice 4	0x0

Table 4-28. WR_LVL_CON2(Address Offset=0x74)

Field	Bit	R/W	Description	Initial Value
ctrl_wrlvl8_code	[7:0]	R/W	Write Level Slave DLL Code Value for Data_Slice 8	0x0

Table 4-29. WR_LVL_CON3(Address Offset=0x78)

Field	Bit	R/W	Description	Initial Value
ctrl_wrlvl_resync	[0]	R/W	Write Level DLL Code Update Enable	0x0

4.11 Command DeSkew Control Register

Table 4-30. CA_DSKEW_CON0(Address Offset=0x7C)

Field	Bit	R/W	Description	Initial Value
CA3DeSkewCode	[31:24]	R/W	DeSkew Code for CA[3] (0x8~0xFF)	0x0
CA2DeSkewCode	[23:16]	R/W	DeSkew Code for CA[2] (0x8~0xFF)	0x0
CA1DeSkewCode	[15:8]	R/W	DeSkew Code for CA[1] (0x8~0xFF)	0x0

Field	Bit	R/W	Description	Initial Value
CA0DeSkewCode	[7:0]	R/W	DeSkew Code for CA[0] (0x8~0xFF)	0x0

Table 4-31. CA_DSKEW_CON1(Address Offset=0x80)

Field	Bit	R/W	Description	Initial Value
CA7DeSkewCode	[31:24]	R/W	DeSkew Code for CA[7] (0x8~0xFF)	0x0
CA6DeSkewCode	[23:16]	R/W	DeSkew Code for CA[6] (0x8~0xFF)	0x0
CA5DeSkewCode	[15:8]	R/W	DeSkew Code for CA[5] (0x8~0xFF)	0x0
CA4DeSkewCode	[7:0]	R/W	DeSkew Code for CA[4] (0x8~0xFF)	0x0

Table 4-32. CA_DSKEW_CON2(Address Offset=0x84)

Field	Bit	R/W	Description	Initial Value
CS0DeSkewCode	[31:24]	R/W	DeSkew Code for CS0 (0x8~0xFF)	0x0
CKDeSkewCode	[23:16]	R/W	DeSkew Code for CK (0x8~0xFF)	0x0
CA9DeSkewCode	[15:8]	R/W	DeSkew Code for CA[9] (0x8~0xFF)	0x0
CA8DeSkewCode	[7:0]	R/W	DeSkew Code for CA[8] (0x8~0xFF)	0x0

Table 4-33. CA_DESKEW_CON3(Address Offset=0x88)

Field	Bit	R/W	Description	Initial Value
CKE1DeSkewCode	[23:16]	R/W	DeSkew Code for CKE1 (0x8~0xFF)	0x0
CKE0DeSkewCode	[15:8]	R/W	DeSkew Code for CAE0 (0x8~0xFF)	0x0
CS1DeSkewCode	[7:0]	R/W	DeSkew Code for CS1 (0x8~0xFF)	0x0

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 4-34. CA_DESKEW_CON4(Address Offset=0x94)

Field	Bit	R/W	Description	Initial Value
RSTDeSkewCode	[7:0]	R/W	DeSkew Code for RST (0x8~0xFF)	0x0

Table 4-35. DRVDS_CON0(Address Offset=0x9C)

Field	Bit	R/W	Description	Initial Value
DS4DrvrDS	[30:28]	R/W	Driver Strength for Data Slice 4	0x0
DS3DrvrDS	[27:25]	R/W	Driver Strength for Data Slice 3	0x0
DS2DrvrDS	[24:22]	R/W	Driver Strength for Data Slice 2	0x0
DS1DrvrDS	[21:19]	R/W	Driver Strength for Data Slice 1	0x0
DS0DrvrDS	[18:16]	R/W	Driver Strength for Data Slice 0	0x0
CaCkDrvrDS	[11:9]	R/W	Driver Strength for CK	0x0
CaCkeDrvrDS	[8:6]	R/W	Driver Strength for Cke[1:0]	0x0
CaCSDrvrDS	[5:3]	R/W	Driver Strength for CS[1:0]	0x0
CaAdrDrvrDS	[2:0]	R/W	Driver Strength for CA[9:0], RAS, CAS, WEN, ODT[1:0], RESET, BANK[2:0].	0x0

Table 4-36. DRVDS_CON1(Address Offset=0xA0)

Field	Bit	R/W	Description	Initial Value
DS8DrvrDS	[11:9]	R/W	Driver Strength for Data Slice 8	0x0
DS7DrvrDS	[8:6]	R/W	Driver Strength for Data Slice 7	0x0
DS6DrvrDS	[5:3]	R/W	Driver Strength for Data Slice 6	0x0
DS5DrvrDS	[2:0]	R/W	Driver Strength for Data Slice 5	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"(=ZQ_CON0[26:24]) will control Driver Strength.

4.12 MDLL Control Register

Table 4-37. MDLL_CON0(Address Offset=0xB0)

Field	Bit	R/W	Description	Initial Value
Reserved	[31]		Should be zero	1'b0
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	1'b0
ctrl_inc	[22:16]	R/W	Increase amount of start point	7'h10

Field	Bit	R/W	Description	Initial Value
ctrl_force	[15:7]	R/W	This field is used instead of ctrl_lock_value[8:0] 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°.	9'h00
ctrl_start	[6]	R/W	This field is used to start DLL locking.	1'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.	1'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	1'b0

Table 4-38. MDLL_CON1(Address Offset=0xB4)

Field	Bit	R/W	Description	Initial Value
ctrl_lock_value	[16:8]	R	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.

4.13 ZQ Control Register

Table 4-39. ZQ_CON0(Address Offset=0xC0)

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
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'h7
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) gDDR3 : 1'b0	1'b0

Field	Bit	R/W	Description	Initial Value
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'h7
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"(=ZQ_CON0[1]) should be toggled after ZQ_CON0[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 4-40. ZQ_CON1(Address Offset=0xC4)

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 4-41. ZQ_CON2(Address Offset=0xC8)

Field	Bit	R/W	Description	Initial Value
ctrl_zq_clk_div	[31:0]	R/W	ZQ Clock(=io_zq_clk) divider setting value. The frequency will be the following formula. "io_zq_clk"(MHz) = clk2x(MHz) / ((ctrl_zq_clk_div+1)*4)	0x7

Table 4-42. ZQ_CON3(Address Offset=0xCC)

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

4.14 Read Data Enable Timing Status Register

Table 4-43. T_RDDATA_CON0(Address Offset=0xD0)

Field	Bit	R/W	Description	Initial Value
T_rddata_en	[28:24]	R/W	This field will be used by Trddata_en timing parameter during Calibration.	0x15
T3_rddata_en	[23:18]	R	Trddata_en timing parameter is read for data slice 3 after Read Calibration.	0x15
T2_rddata_en	[17:12]	R	Trddata_en timing parameter is read for data slice 2 after Read Calibration.	0x15
T1_rddata_en	[11:6]	R	Trddata_en timing parameter for data slice 1 after Read Calibration.	0x15
T0_rddata_en	[5:0]	R	Trddata_en timing parameter for data slice 0 after Read Calibration.	0x15

Table 4-44. T_RDDATA_CON1(Address Offset=0xD4)

Field	Bit	R/W	Description	Initial Value
T7_rddata_en	[23:18]	R	Trddata_en timing parameter is read for data slice 3 after Read Calibration.	0x15
T6_rddata_en	[17:12]	R	Trddata_en timing parameter is read for data slice 2 after Read Calibration.	0x15
T5_rddata_en	[11:6]	R	Trddata_en timing parameter for data slice 1 after Read Calibration.	0x15
T4_rddata_en	[5:0]	R	Trddata_en timing parameter for data slice 0 after Read Calibration.	0x15

Table 4-45. T_RDDATA_CON2(Address Offset=0xD8)

Field	Bit	R/W	Description	Initial Value
T8_rddata_en	[5:0]	R	Trddata_en timing parameter for data slice 0 after Read Calibration.	0x15

4.15 Calibration Status Register

Table 4-46. CAL_WL_STAT(Address Offset=0xDC)

Field	Bit	R/W	Description	Initial Value
wl_cal_status	[8:0]	R	It will be disabled if there is a fail status after WL Calibration. It should be read by all one if Calibration is done normally.	0x0

Table 4-47. CAL_FAIL_STAT0(Address Offset=0xE0)

Field	Bit	R/W	Description	Initial Value
dq_fail_status	[31:24]	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. (slice3)	0x0
dq_fail_status	[23:16]	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. (slice2)	0x0
dq_fail_status	[15:8]	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. (slice1)	0x0
dq_fail_status	[7:0]	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. (slice0)	0x0

Table 4-48. CAL_FAIL_STAT1(Address Offset=0xE4)

Field	Bit	R/W	Description	Initial Value
dq_fail_status	[31:24]	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. (slice7)	0x0
dq_fail_status	[23:16]	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. (slice6)	0x0
dq_fail_status	[15:8]	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. (slice5)	0x0
dq_fail_status	[7:0]	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. (slice4)	0x0

Table 4-49. CAL_FAIL_STAT2(Address Offset=0xE8)

Field	Bit	R/W	Description	Initial Value
dq_fail_status	[7:0]	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. (slice8)	0x0

Table 4-50. CAL_FAIL_STAT3(Address Offset=0xEC)

Field	Bit	R/W	Description	Initial Value
dm_fail_status	[8:0]	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.	0x0

4.16 Valid Window Margin Register

Table 4-51. CAL_GT_VWMC0(Address Offset=0xF0)

Field	Bit	R/W	Description	Initial Value
GT_VWMC3	[31:24]	R	Gate Training Centering code for data slice 3.	-
GT_VWMC2	[23:16]	R	Gate Training Centering code for data slice 2.	-
GT_VWMC1	[15:8]	R	Gate Training Centering code for data slice 1.	-
GT_VWMC0	[7:0]	R	Gate Training Centering code for data slice 0.	-

Table 4-52. CAL_GT_VWMC1(Address Offset=0xF4)

Field	Bit	R/W	Description	Initial Value
GT_VWMC3	[31:24]	R	Gate Training Centering code for data slice 7.	-
GT_VWMC2	[23:16]	R	Gate Training Centering code for data slice 6.	-
GT_VWMC1	[15:8]	R	Gate Training Centering code for data slice 5.	-
GT_VWMC0	[7:0]	R	Gate Training Centering code for data slice 4.	-

Table 4-53. CAL_GT_VWMC2(Address Offset=0xF8)

Field	Bit	R/W	Description	Initial Value
GT_VWMC8	[7:0]	R	Gate Training Centering code for data slice 8.	-

Table 4-54. CAL_GT_CYC(Address Offset=0xFC)

Field	Bit	R/W	Description	Initial Value
GT_CYC8	[26:24]	R	Gate Training Centering code for data slice 8.	-
GT_CYC7	[23:21]	R	Gate Training Centering code for data slice 7.	-
GT_CYC6	[20:18]	R	Gate Training Centering code for data slice 6.	-
GT_CYC5	[17:15]	R	Gate Training Centering code for data slice 5.	-
GT_CYC4	[14:12]	R	Gate Training Centering code for data slice 4.	-
GT_CYC3	[11:9]	R	Gate Training Centering code for data slice 3.	-

Field	Bit	R/W	Description	Initial Value
GT_CYC2	[8:6]	R	Gate Training Centering code for data slice 2.	-
GT_CYC1	[5:3]	R	Gate Training Centering code for data slice 1.	-
GT_CYC0	[2:0]	R	Gate Training Centering code for data slice 0.	-

Table 4-55. CAL_RD_VWMC0(Address Offset=0x100)

Field	Bit	R/W	Description	Initial Value
rd_vwmc3	[31:24]	R	DQ Calibration centering code for data slice 3.	-
rd_vwmc2	[23:16]	R	DQ Calibration centering code for data slice 2.	-
rd_vwmc1	[15:8]	R	DQ Calibration centering code for data slice 1.	-
rd_vwmc0	[7:0]	R	DQ Calibration centering code for data slice 0.	-

Table 4-56. CAL_RD_VWMC1(Address Offset=0x104)

Field	Bit	R/W	Description	Initial Value
rd_vwmc7	[31:24]	R	DQ Calibration centering code for data slice 7.	-
rd_vwmc6	[23:16]	R	DQ Calibration centering code for data slice 6.	-
rd_vwmc5	[15:8]	R	DQ Calibration centering code for data slice 5.	-
rd_vwmc4	[7:0]	R	DQ Calibration centering code for data slice 4.	-

Table 4-57. CAL_RD_VWMC2(Address Offset=0x108)

Field	Bit	R/W	Description	Initial Value
rd_vwmc8	[7:0]	R	DQ Calibration centering code for data slice 8.	-

Table 4-58. CAL_RD_VWML0(Address Offset=0x110)

Field	Bit	R/W	Description	Initial Value
rd_vwml3	[31:24]	R	Left Code Value in Read Valid Window Margin for Data Slice3.	-
rd_vwml2	[23:16]	R	Left Code Value in Read Valid Window Margin for Data Slice2.	-
rd_vwml1	[15:8]	R	Left Code Value in Read Valid Window Margin for Data Slice1.	-
rd_vwml0	[7:0]	R	Left Code Value in Read Valid Window Margin for Data Slice0.	-

Table 4-59. CAL_RD_VWML1(Address Offset=0x114)

Field	Bit	R/W	Description	Initial Value

Field	Bit	R/W	Description	Initial Value
rd_vwml7	[31:24]	R	Left Code Value in Read Valid Window Margin for Data Slice7.	-
rd_vwml6	[23:16]	R	Left Code Value in Read Valid Window Margin for Data Slice6.	-
rd_vwml5	[15:8]	R	Left Code Value in Read Valid Window Margin for Data Slice5.	-
rd_vwml4	[7:0]	R	Left Code Value in Read Valid Window Margin for Data Slice4.	-

Table 4-60. CAL_RD_VWML2(Address Offset=0x118)

Field	Bit	R/W	Description	Initial Value
rd_vwml8	[7:0]	R	Left Code Value in Read Valid Window Margin for Data Slice8.	-

Table 4-61. CAL_RD_VWMR0(Address Offset=0x120)

Field	Bit	R/W	Description	Initial Value
rd_vwmr3	[31:24]	R	Right Code Value in Read Valid Window Margin for Data Slice3.	-
rd_vwmr2	[23:16]	R	Right Code Value in Read Valid Window Margin for Data Slice2.	-
rd_vwmr1	[15:8]	R	Right Code Value in Read Valid Window Margin for Data Slice1.	-
rd_vwmr0	[7:0]	R	Right Code Value in Read Valid Window Margin for Data Slice0.	-

Table 4-62. CAL_RD_VWMR1(Address Offset=0x124)

Field	Bit	R/W	Description	Initial Value
rd_vwmr7	[31:24]	R	Right Code Value in Read Valid Window Margin for Data Slice7.	-
rd_vwmr6	[23:16]	R	Right Code Value in Read Valid Window Margin for Data Slice6.	-
rd_vwmr5	[15:8]	R	Right Code Value in Read Valid Window Margin for Data Slice5.	-
rd_vwmr4	[7:0]	R	Right Code Value in Read Valid Window Margin for Data Slice4.	-

Table 4-63. CAL_RD_VWMR2(Address Offset=0x128)

Field	Bit	R/W	Description	Initial Value
rd_vwmr8	[7:0]	R	Right Code Value in Read Valid Window Margin for Data Slice8.	-

Table 4-64. CAL_WR_VWMC0(Address Offset=0x130)

Field	Bit	R/W	Description	Initial Value
wr_vwmc3	[31:24]	R	DQ Calibration centering code for data slice 3.	-
wr_vwmc2	[23:16]	R	DQ Calibration centering code for data slice 2.	-

Field	Bit	R/W	Description	Initial Value
wr_vwmc1	[15:8]	R	DQ Calibration centering code for data slice 1.	-
wr_vwmc0	[7:0]	R	DQ Calibration centering code for data slice 0.	-

Table 4-65. CAL_WR_VWMC1(Address Offset=0x134)

Field	Bit	R/W	Description	Initial Value
wr_vwmc7	[31:24]	R	DQ Calibration centering code for data slice 7.	-
wr_vwmc6	[23:16]	R	DQ Calibration centering code for data slice 6.	-
wr_vwmc5	[15:8]	R	DQ Calibration centering code for data slice 5.	-
wr_vwmc4	[7:0]	R	DQ Calibration centering code for data slice 4.	-

Table 4-66. CAL_WR_VWMC2(Address Offset=0x138)

Field	Bit	R/W	Description	Initial Value
wr_vwmc8	[7:0]	R	DQ Calibration centering code for data slice 8.	-

Table 4-67. CAL_WR_VWML0(Address Offset=0x140)

Field	Bit	R/W	Description	Initial Value
wr_vwml3	[31:24]	R	Left Code Value in Write Valid Window Margin for Data Slice3.	-
wr_vwml2	[23:16]	R	Left Code Value in Write Valid Window Margin for Data Slice2.	-
wr_vwml1	[15:8]	R	Left Code Value in Write Valid Window Margin for Data Slice1.	-
wr_vwml0	[7:0]	R	Left Code Value in Write Valid Window Margin for Data Slice0.	-

Table 4-68. CAL_WR_VWML1(Address Offset=0x144)

Field	Bit	R/W	Description	Initial Value
wr_vwml7	[31:24]	R	Left Code Value in Write Valid Window Margin for Data Slice7.	-
wr_vwml6	[23:16]	R	Left Code Value in Write Valid Window Margin for Data Slice6.	-
wr_vwml5	[15:8]	R	Left Code Value in Write Valid Window Margin for Data Slice5.	-
wr_vwml4	[7:0]	R	Left Code Value in Write Valid Window Margin for Data Slice4.	-

Table 4-69. CAL_WR_VWML2(Address Offset=0x148)

Field	Bit	R/W	Description	Initial Value
wr_vwml8	[7:0]	R	Left Code Value in Write Valid Window Margin for Data Slice8.	-

Table 4-70. CAL_WR_VWMR0(Address Offset=0x150)

Field	Bit	R/W	Description	Initial Value
wr_vwmr3	[31:24]	R	Right Code Value in Write Valid Window Margin for Data Slice3.	-
wr_vwmr2	[23:16]	R	Right Code Value in Write Valid Window Margin for Data Slice2.	-
wr_vwmr1	[15:8]	R	Right Code Value in Write Valid Window Margin for Data Slice1.	-
wr_vwmr0	[7:0]	R	Right Code Value in Write Valid Window Margin for Data Slice0.	-

Table 4-71. CAL_WR_VWMR1(Address Offset=0x154)

Field	Bit	R/W	Description	Initial Value
wr_vwmr7	[31:24]	R	Right Code Value in Write Valid Window Margin for Data Slice7.	-
wr_vwmr6	[23:16]	R	Right Code Value in Write Valid Window Margin for Data Slice6.	-
wr_vwmr5	[15:8]	R	Right Code Value in Write Valid Window Margin for Data Slice5.	-
wr_vwmr4	[7:0]	R	Right Code Value in Write Valid Window Margin for Data Slice4.	-

Table 4-72. CAL_WR_VWMR2(Address Offset=0x158)

Field	Bit	R/W	Description	Initial Value
wr_vwmr8	[7:0]	R	Right Code Value in Write Valid Window Margin for Data Slice8.	-

Table 4-73. CAL_DM_VWMC0(Address Offset=0x160)

Field	Bit	R/W	Description	Initial Value
dm_vwmc3	[31:24]	R	DQ Calibration centering code for data slice 3.	-
dm_vwmc2	[23:16]	R	DQ Calibration centering code for data slice 2.	-
dm_vwmc1	[15:8]	R	DQ Calibration centering code for data slice 1.	-
dm_vwmc0	[7:0]	R	DQ Calibration centering code for data slice 0.	-

Table 4-74. CAL_DM_VWMC1(Address Offset=0x164)

Field	Bit	R/W	Description	Initial Value
dm_vwmc7	[31:24]	R	DQ Calibration centering code for data slice 7.	-
dm_vwmc6	[23:16]	R	DQ Calibration centering code for data slice 6.	-
dm_vwmc5	[15:8]	R	DQ Calibration centering code for data slice 5.	-
dm_vwmc4	[7:0]	R	DQ Calibration centering code for data slice 4.	-

Table 4-75. CAL_DM_VWMC2(Address Offset=0x168)

Field	Bit	R/W	Description	Initial Value
dm_vwmc8	[7:0]	R	DQ Calibration centering code for data slice 8.	-

Table 4-76. CAL_DM_VWML0(Address Offset=0x170)

Field	Bit	R/W	Description	Initial Value
dm_vwml3	[31:24]	R	Left Code Value in DM Valid Window Margin for Data Slice3.	-
dm_vwml2	[23:16]	R	Left Code Value in DM Valid Window Margin for Data Slice2.	-
dm_vwml1	[15:8]	R	Left Code Value in DM Valid Window Margin for Data Slice1.	-
dm_vwml0	[7:0]	R	Left Code Value in DM Valid Window Margin for Data Slice0.	-

Table 4-77. CAL_DM_VWML1(Address Offset=0x174)

Field	Bit	R/W	Description	Initial Value
dm_vwml7	[31:24]	R	Left Code Value in DM Valid Window Margin for Data Slice7.	-
dm_vwml6	[23:16]	R	Left Code Value in DM Valid Window Margin for Data Slice6.	-
dm_vwml5	[15:8]	R	Left Code Value in DM Valid Window Margin for Data Slice5.	-
dm_vwml4	[7:0]	R	Left Code Value in DM Valid Window Margin for Data Slice4.	-

Table 4-78. CAL_DM_VWML2(Address Offset=0x178)

Field	Bit	R/W	Description	Initial Value
dm_vwml8	[7:0]	R	Left Code Value in DM Valid Window Margin for Data Slice8.	-

Table 4-79. CAL_DM_VWMR0(Address Offset=0x180)

Field	Bit	R/W	Description	Initial Value
dm_vwmr3	[31:24]	R	Right Code Value in DM Valid Window Margin for Data Slice3.	-
dm_vwmr2	[23:16]	R	Right Code Value in DM Valid Window Margin for Data Slice2.	-
dm_vwmr1	[15:8]	R	Right Code Value in DM Valid Window Margin for Data Slice1.	-
dm_vwmr0	[7:0]	R	Right Code Value in DM Valid Window Margin for Data Slice0.	-

Table 4-80. CAL_DM_VWMR1(Address Offset=0x184)

Field	Bit	R/W	Description	Initial Value
dm_vwmr7	[31:24]	R	Right Code Value in DM Valid Window Margin for Data Slice7.	-

Field	Bit	R/W	Description	Initial Value
dm_vwmr6	[23:16]	R	Right Code Value in DM Valid Window Margin for Data Slice6.	-
dm_vwmr5	[15:8]	R	Right Code Value in DM Valid Window Margin for Data Slice5.	-
dm_vwmr4	[7:0]	R	Right Code Value in DM Valid Window Margin for Data Slice4.	-

Table 4-81. CAL_DM_VWMR2(Address Offset=0x188)

Field	Bit	R/W	Description	Initial Value
dm_vwmr8	[7:0]	R	Right Code Value in DM Valid Window Margin for Data Slice8.	-

4.17 Read De-skew Control Register

Table 4-82. RD_DESKEW_CON0(Address Offset=0x190)

Field	Bit	R/W	Description	Initial Value
RD0Deskew3	[31:24]	R	Read DQ0 De-Skew Code for Data Slice3.	-
RD0Deskew2	[23:16]	R	Read DQ0 De-Skew Code for Data Slice2.	-
RD0Deskew1	[15:8]	R	Read DQ0 De-Skew Code for Data Slice1.	-
RD0Deskew0	[7:0]	R	Read DQ0 De-Skew Code for Data Slice0.	-

Table 4-83. RD_DESKEW_CON1(Address Offset=0x194)

Field	Bit	R/W	Description	Initial Value
RD0Deskew7	[31:24]	R	Read DQ0 De-Skew Code for Data Slice7.	-
RD0Deskew6	[23:16]	R	Read DQ0 De-Skew Code for Data Slice6.	-
RD0Deskew5	[15:8]	R	Read DQ0 De-Skew Code for Data Slice5.	-
RD0Deskew4	[7:0]	R	Read DQ0 De-Skew Code for Data Slice4.	-

Table 4-84. RD_DESKEW_CON2(Address Offset=0x198)

Field	Bit	R/W	Description	Initial Value
RD0Deskew8	[7:0]	R	Read DQ0 De-Skew Code for Data Slice8.	-

Table 4-85. RD_DESKEW_CON3(Address Offset=0x19C)

Field	Bit	R/W	Description	Initial Value

Field	Bit	R/W	Description	Initial Value
RD1Deskew3	[31:24]	R	Read DQ1 De-Skew Code for Data Slice3.	-
RD1Deskew2	[23:16]	R	Read DQ1 De-Skew Code for Data Slice2.	-
RD1Deskew1	[15:8]	R	Read DQ1 De-Skew Code for Data Slice1.	-
RD1Deskew0	[7:0]	R	Read DQ1 De-Skew Code for Data Slice0.	-

Table 4-86. RD_DESKEW_CON4(Address Offset=0x1A0)

Field	Bit	R/W	Description	Initial Value
RD1Deskew7	[31:24]	R	Read DQ1 De-Skew Code for Data Slice7.	-
RD1Deskew6	[23:16]	R	Read DQ1 De-Skew Code for Data Slice6.	-
RD1Deskew5	[15:8]	R	Read DQ1 De-Skew Code for Data Slice5.	-
RD1Deskew4	[7:0]	R	Read DQ1 De-Skew Code for Data Slice4.	-

Table 4-87. RD_DESKEW_CON5(Address Offset=0x1A4)

Field	Bit	R/W	Description	Initial Value
RD1Deskew8	[7:0]	R	Read DQ1 De-Skew Code for Data Slice8.	-

Table 4-88. RD_DESKEW_CON6(Address Offset=0x1A8)

Field	Bit	R/W	Description	Initial Value
RD2Deskew3	[31:24]	R	Read DQ2 De-Skew Code for Data Slice3.	-
RD2Deskew2	[23:16]	R	Read DQ2 De-Skew Code for Data Slice2.	-
RD2Deskew1	[15:8]	R	Read DQ2 De-Skew Code for Data Slice1.	-
RD2Deskew0	[7:0]	R	Read DQ2 De-Skew Code for Data Slice0.	-

Table 4-89. RD_DESKEW_CON7(Address Offset=0x1AC)

Field	Bit	R/W	Description	Initial Value
RD2Deskew7	[31:24]	R	Read DQ2 De-Skew Code for Data Slice7.	-
RD2Deskew6	[23:16]	R	Read DQ2 De-Skew Code for Data Slice6.	-
RD2Deskew5	[15:8]	R	Read DQ2 De-Skew Code for Data Slice5.	-
RD2Deskew4	[7:0]	R	Read DQ2 De-Skew Code for Data Slice4.	-

Table 4-90. RD_DESKEW_CON8(Address Offset=0x1B0)

Field	Bit	R/W	Description	Initial Value
RD2Deskew8	[7:0]	R	Read DQ2 De-Skew Code for Data Slice8.	-

Table 4-91. RD_DESKEW_CON9(Address Offset=0x1B4)

Field	Bit	R/W	Description	Initial Value
RD3Deskew3	[31:24]	R	Read DQ3 De-Skew Code for Data Slice3.	-
RD3Deskew2	[23:16]	R	Read DQ3 De-Skew Code for Data Slice2.	-
RD3Deskew1	[15:8]	R	Read DQ3 De-Skew Code for Data Slice1.	-
RD3Deskew0	[7:0]	R	Read DQ3 De-Skew Code for Data Slice0.	-

Table 4-92. RD_DESKEW_CON10(Address Offset=0x1B8)

Field	Bit	R/W	Description	Initial Value
RD3Deskew7	[31:24]	R	Read DQ3 De-Skew Code for Data Slice7.	-
RD3Deskew6	[23:16]	R	Read DQ3 De-Skew Code for Data Slice6.	-
RD3Deskew5	[15:8]	R	Read DQ3 De-Skew Code for Data Slice5.	-
RD3Deskew4	[7:0]	R	Read DQ3 De-Skew Code for Data Slice4.	-

Table 4-93. RD_DESKEW_CON11(Address Offset=0x1BC)

Field	Bit	R/W	Description	Initial Value
RD3Deskew8	[7:0]	R	Read DQ3 De-Skew Code for Data Slice8.	-

Table 4-94. RD_DESKEW_CON12(Address Offset=0x1C0)

Field	Bit	R/W	Description	Initial Value
RD4Deskew3	[31:24]	R	Read DQ4 De-Skew Code for Data Slice3.	-
RD4Deskew2	[23:16]	R	Read DQ4 De-Skew Code for Data Slice2.	-
RD4Deskew1	[15:8]	R	Read DQ4 De-Skew Code for Data Slice1.	-
RD4Deskew0	[7:0]	R	Read DQ4 De-Skew Code for Data Slice0.	-

Table 4-95. RD_DESKEW_CON13(Address Offset=0x1C4)

Field	Bit	R/W	Description	Initial Value
RD4Deskew7	[31:24]	R	Read DQ4 De-Skew Code for Data Slice7.	-
RD4Deskew6	[23:16]	R	Read DQ4 De-Skew Code for Data Slice6.	-

Field	Bit	R/W	Description	Initial Value
RD4Deskew5	[15:8]	R	Read DQ4 De-Skew Code for Data Slice5.	-
RD4Deskew4	[7:0]	R	Read DQ4 De-Skew Code for Data Slice4.	-

Table 4-96. RD_DESKEW_CON14(Address Offset=0x1C8)

Field	Bit	R/W	Description	Initial Value
RD4Deskew8	[7:0]	R	Read DQ4 De-Skew Code for Data Slice8.	-

Table 4-97. RD_DESKEW_CON15(Address Offset=0x1CC)

Field	Bit	R/W	Description	Initial Value
RD5Deskew3	[31:24]	R	Read DQ5 De-Skew Code for Data Slice3.	-
RD5Deskew2	[23:16]	R	Read DQ5 De-Skew Code for Data Slice2.	-
RD5Deskew1	[15:8]	R	Read DQ5 De-Skew Code for Data Slice1.	-
RD5Deskew0	[7:0]	R	Read DQ5 De-Skew Code for Data Slice0.	-

Table 4-98. RD_DESKEW_CON16(Address Offset=0x1D0)

Field	Bit	R/W	Description	Initial Value
RD5Deskew7	[31:24]	R	Read DQ5 De-Skew Code for Data Slice7.	-
RD5Deskew6	[23:16]	R	Read DQ5 De-Skew Code for Data Slice6.	-
RD5Deskew5	[15:8]	R	Read DQ5 De-Skew Code for Data Slice5.	-
RD5Deskew4	[7:0]	R	Read DQ5 De-Skew Code for Data Slice4.	-

Table 4-99. RD_DESKEW_CON17(Address Offset=0x1D4)

Field	Bit	R/W	Description	Initial Value
RD5Deskew8	[7:0]	R	Read DQ5 De-Skew Code for Data Slice8.	-

Table 4-100. RD_DESKEW_CON18(Address Offset=0x1D8)

Field	Bit	R/W	Description	Initial Value
RD6Deskew3	[31:24]	R	Read DQ6 De-Skew Code for Data Slice3.	-
RD6Deskew2	[23:16]	R	Read DQ6 De-Skew Code for Data Slice2.	-
RD6Deskew1	[15:8]	R	Read DQ6 De-Skew Code for Data Slice1.	-
RD6Deskew0	[7:0]	R	Read DQ6 De-Skew Code for Data Slice0.	-

Table 4-101. RD_DESKEW_CON19(Address Offset=0x1DC)

Field	Bit	R/W	Description	Initial Value
RD6Deskew7	[31:24]	R	Read DQ6 De-Skew Code for Data Slice7.	-
RD6Deskew6	[23:16]	R	Read DQ6 De-Skew Code for Data Slice6.	-
RD6Deskew5	[15:8]	R	Read DQ6 De-Skew Code for Data Slice5.	-
RD6Deskew4	[7:0]	R	Read DQ6 De-Skew Code for Data Slice4.	-

Table 4-102. RD_DESKEW_CON20(Address Offset=0x1E0)

Field	Bit	R/W	Description	Initial Value
RD6Deskew8	[7:0]	R	Read DQ6 De-Skew Code for Data Slice8.	-

Table 4-103. RD_DESKEW_CON21(Address Offset=0x1E4)

Field	Bit	R/W	Description	Initial Value
RD7Deskew3	[31:24]	R	Read DQ7 De-Skew Code for Data Slice3.	-
RD7Deskew2	[23:16]	R	Read DQ7 De-Skew Code for Data Slice2.	-
RD7Deskew1	[15:8]	R	Read DQ7 De-Skew Code for Data Slice1.	-
RD7Deskew0	[7:0]	R	Read DQ7 De-Skew Code for Data Slice0.	-

Table 4-104. RD_DESKEW_CON22(Address Offset=0x1E8)

Field	Bit	R/W	Description	Initial Value
RD7Deskew7	[31:24]	R	Read DQ7 De-Skew Code for Data Slice7.	-
RD7Deskew6	[23:16]	R	Read DQ7 De-Skew Code for Data Slice6.	-
RD7Deskew5	[15:8]	R	Read DQ7 De-Skew Code for Data Slice5.	-
RD7Deskew4	[7:0]	R	Read DQ7 De-Skew Code for Data Slice4.	-

Table 4-105. RD_DESKEW_CON23(Address Offset=0x1EC)

Field	Bit	R/W	Description	Initial Value
RD7Deskew8	[7:0]	R	Read DQ7 De-Skew Code for Data Slice8.	-

- ~~According to Reg_mode[7:0] and RdDeskew_clear, "ctrl_offsetr*"(=OFFSETR_CON0,1,2) will be written to "Read Deskew Code" register instead of "ctrl_offsetr*" itself.~~

  ~~Reg_mode[7:0]=8'h02 : DQ0 is written by "ctrl_offsetr*".~~

- ~~Reg_mode[7:0]=8'h12 : DQ1 is written by "ctrl_offsetr*".~~
- ~~Reg_mode[7:0]=8'h22 : DQ2 is written by "ctrl_offsetr*".~~
- ~~Reg_mode[7:0]=8'h32 : DQ3 is written by "ctrl_offsetr*".~~
- ~~Reg_mode[7:0]=8'h42 : DQ4 is written by "ctrl_offsetr*".~~
- ~~Reg_mode[7:0]=8'h52 : DQ5 is written by "ctrl_offsetr*".~~
- ~~Reg_mode[7:0]=8'h62 : DQ6 is written by "ctrl_offsetr*".~~
- ~~Reg_mode[7:0]=8'h72 : DQ7 is written by "ctrl_offsetr*".~~
- ~~Reg_mode[7:0]=8'h82 : All DQ Deskew Codes is cleared.~~

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

4.18 Write De-skew Control Register

Table 4-106. WR_DESKEW_CON0(Address Offset=0x1F0)

Field	Bit	R/W	Description	Initial Value
WR0Deskew3	[31:24]	R	Write DQ0 De-Skew Code for Data Slice3.	-
WR0Deskew2	[23:16]	R	Write DQ0 De-Skew Code for Data Slice2.	-
WR0Deskew1	[15:8]	R	Write DQ0 De-Skew Code for Data Slice1.	-
WR0Deskew0	[7:0]	R	Write DQ0 De-Skew Code for Data Slice0.	-

Table 4-107. WR_DESKEW_CON1(Address Offset=0x1F4)

Field	Bit	R/W	Description	Initial Value
WR0Deskew7	[31:24]	R	Write DQ0 De-Skew Code for Data Slice7.	-
WR0Deskew6	[23:16]	R	Write DQ0 De-Skew Code for Data Slice6.	-
WR0Deskew5	[15:8]	R	Write DQ0 De-Skew Code for Data Slice5.	-
WR0Deskew4	[7:0]	R	Write DQ0 De-Skew Code for Data Slice4.	-

Table 4-108. WR_DESKEW_CON2(Address Offset=0x1F8)

Field	Bit	R/W	Description	Initial Value
WR0Deskew8	[7:0]	R	Write DQ0 De-Skew Code for Data Slice8.	-

Table 4-109. WR_DESKEW_CON3(Address Offset=0x1FC)

Field	Bit	R/W	Description	Initial Value
WR1Deskew3	[31:24]	R	Write DQ1 De-Skew Code for Data Slice3.	-
WR1Deskew2	[23:16]	R	Write DQ1 De-Skew Code for Data Slice2.	-

Field	Bit	R/W	Description	Initial Value
WR1Deskew1	[15:8]	R	Write DQ1 De-Skew Code for Data Slice1.	-
WR1Deskew0	[7:0]	R	Write DQ1 De-Skew Code for Data Slice0.	-

Table 4-110. WR_DESKEW_CON4(Address Offset=0x200)

Field	Bit	R/W	Description	Initial Value
WR1Deskew7	[31:24]	R	Write DQ1 De-Skew Code for Data Slice7.	-
WR1Deskew6	[23:16]	R	Write DQ1 De-Skew Code for Data Slice6.	-
WR1Deskew5	[15:8]	R	Write DQ1 De-Skew Code for Data Slice5.	-
WR1Deskew4	[7:0]	R	Write DQ1 De-Skew Code for Data Slice4.	-

Table 4-111. WR_DESKEW_CON5(Address Offset=0x204)

Field	Bit	R/W	Description	Initial Value
WR1Deskew8	[7:0]	R	Write DQ1 De-Skew Code for Data Slice8.	-

Table 4-112. WR_DESKEW_CON6(Address Offset=0x208)

Field	Bit	R/W	Description	Initial Value
WR2Deskew3	[31:24]	R	Write DQ2 De-Skew Code for Data Slice3.	-
WR2Deskew2	[23:16]	R	Write DQ2 De-Skew Code for Data Slice2.	-
WR2Deskew1	[15:8]	R	Write DQ2 De-Skew Code for Data Slice1.	-
WR2Deskew0	[7:0]	R	Write DQ2 De-Skew Code for Data Slice0.	-

Table 4-113. WR_DESKEW_CON7(Address Offset=0x20C)

Field	Bit	R/W	Description	Initial Value
WR2Deskew7	[31:24]	R	Write DQ2 De-Skew Code for Data Slice7.	-
WR2Deskew6	[23:16]	R	Write DQ2 De-Skew Code for Data Slice6.	-
WR2Deskew5	[15:8]	R	Write DQ2 De-Skew Code for Data Slice5.	-
WR2Deskew4	[7:0]	R	Write DQ2 De-Skew Code for Data Slice4.	-

Table 4-114. WR_DESKEW_CON8(Address Offset=0x210)

Field	Bit	R/W	Description	Initial Value
WR2Deskew8	[7:0]	R	Write DQ2 De-Skew Code for Data Slice8.	-

Table 4-115. WR_DESKEW_CON9(Address Offset=0x214)

Field	Bit	R/W	Description	Initial Value
WR3Deskew3	[31:24]	R	Write DQ3 De-Skew Code for Data Slice3.	-
WR3Deskew2	[23:16]	R	Write DQ3 De-Skew Code for Data Slice2.	-
WR3Deskew1	[15:8]	R	Write DQ3 De-Skew Code for Data Slice1.	-
WR3Deskew0	[7:0]	R	Write DQ3 De-Skew Code for Data Slice0.	-

Table 4-116. WR_DESKEW_CON10(Address Offset=0x218)

Field	Bit	R/W	Description	Initial Value
WR3Deskew7	[31:24]	R	Write DQ3 De-Skew Code for Data Slice7.	-
WR3Deskew6	[23:16]	R	Write DQ3 De-Skew Code for Data Slice6.	-
WR3Deskew5	[15:8]	R	Write DQ3 De-Skew Code for Data Slice5.	-
WR3Deskew4	[7:0]	R	Write DQ3 De-Skew Code for Data Slice4.	-

Table 4-117. WR_DESKEW_CON11(Address Offset=0x21C)

Field	Bit	R/W	Description	Initial Value
WR3Deskew8	[7:0]	R	Write DQ3 De-Skew Code for Data Slice8.	-

Table 4-118. WR_DESKEW_CON12(Address Offset=0x220)

Field	Bit	R/W	Description	Initial Value
WR4Deskew3	[31:24]	R	Write DQ4 De-Skew Code for Data Slice3.	-
WR4Deskew2	[23:16]	R	Write DQ4 De-Skew Code for Data Slice2.	-
WR4Deskew1	[15:8]	R	Write DQ4 De-Skew Code for Data Slice1.	-
WR4Deskew0	[7:0]	R	Write DQ4 De-Skew Code for Data Slice0.	-

Table 4-119. WR_DESKEW_CON13(Address Offset=0x224)

Field	Bit	R/W	Description	Initial Value
WR4Deskew7	[31:24]	R	Write DQ4 De-Skew Code for Data Slice7.	-
WR4Deskew6	[23:16]	R	Write DQ4 De-Skew Code for Data Slice6.	-
WR4Deskew5	[15:8]	R	Write DQ4 De-Skew Code for Data Slice5.	-
WR4Deskew4	[7:0]	R	Write DQ4 De-Skew Code for Data Slice4.	-

Table 4-120. WR_DESKEW_CON14(Address Offset=0x228)

Field	Bit	R/W	Description	Initial Value
WR4Deskew8	[7:0]	R	Write DQ4 De-Skew Code for Data Slice8.	-

Table 4-121. WR_DESKEW_CON15(Address Offset=0x22C)

Field	Bit	R/W	Description	Initial Value
WR5Deskew3	[31:24]	R	Write DQ5 De-Skew Code for Data Slice3.	-
WR5Deskew2	[23:16]	R	Write DQ5 De-Skew Code for Data Slice2.	-
WR5Deskew1	[15:8]	R	Write DQ5 De-Skew Code for Data Slice1.	-
WR5Deskew0	[7:0]	R	Write DQ5 De-Skew Code for Data Slice0.	-

Table 4-122. WR_DESKEW_CON16(Address Offset=0x230)

Field	Bit	R/W	Description	Initial Value
WR5Deskew7	[31:24]	R	Write DQ5 De-Skew Code for Data Slice7.	-
WR5Deskew6	[23:16]	R	Write DQ5 De-Skew Code for Data Slice6.	-
WR5Deskew5	[15:8]	R	Write DQ5 De-Skew Code for Data Slice5.	-
WR5Deskew4	[7:0]	R	Write DQ5 De-Skew Code for Data Slice4.	-

Table 4-123. WR_DESKEW_CON17(Address Offset=0x234)

Field	Bit	R/W	Description	Initial Value
WR5Deskew8	[7:0]	R	Write DQ5 De-Skew Code for Data Slice8.	-

Table 4-124. WR_DESKEW_CON18(Address Offset=0x238)

Field	Bit	R/W	Description	Initial Value
WR6Deskew3	[31:24]	R	Write DQ6 De-Skew Code for Data Slice3.	-
WR6Deskew2	[23:16]	R	Write DQ6 De-Skew Code for Data Slice2.	-
WR6Deskew1	[15:8]	R	Write DQ6 De-Skew Code for Data Slice1.	-
WR6Deskew0	[7:0]	R	Write DQ6 De-Skew Code for Data Slice0.	-

Table 4-125. WR_DESKEW_CON19(Address Offset=0x23C)

Field	Bit	R/W	Description	Initial Value
WR6Deskew7	[31:24]	R	Write DQ6 De-Skew Code for Data Slice7.	-

Field	Bit	R/W	Description	Initial Value
WR6Deskew6	[23:16]	R	Write DQ6 De-Skew Code for Data Slice6.	-
WR6Deskew5	[15:8]	R	Write DQ6 De-Skew Code for Data Slice5.	-
WR6Deskew4	[7:0]	R	Write DQ6 De-Skew Code for Data Slice4.	-

Table 4-126. WR_DESKEW_CON20(Address Offset=0x240)

Field	Bit	R/W	Description	Initial Value
WR6Deskew8	[7:0]	R	Write DQ6 De-Skew Code for Data Slice8.	-

Table 4-127. WR_DESKEW_CON21(Address Offset=0x244)

Field	Bit	R/W	Description	Initial Value
WR7Deskew3	[31:24]	R	Write DQ7 De-Skew Code for Data Slice3.	-
WR7Deskew2	[23:16]	R	Write DQ7 De-Skew Code for Data Slice2.	-
WR7Deskew1	[15:8]	R	Write DQ7 De-Skew Code for Data Slice1.	-
WR7Deskew0	[7:0]	R	Write DQ7 De-Skew Code for Data Slice0.	-

Table 4-128. WR_DESKEW_CON22(Address Offset=0x248)

Field	Bit	R/W	Description	Initial Value
WR7Deskew7	[31:24]	R	Write DQ7 De-Skew Code for Data Slice7.	-
WR7Deskew6	[23:16]	R	Write DQ7 De-Skew Code for Data Slice6.	-
WR7Deskew5	[15:8]	R	Write DQ7 De-Skew Code for Data Slice5.	-
WR7Deskew4	[7:0]	R	Write DQ7 De-Skew Code for Data Slice4.	-

Table 4-129. WR_DESKEW_CON23(Address Offset=0x24C)

Field	Bit	R/W	Description	Initial Value
WR7Deskew8	[7:0]	R	Write DQ7 De-Skew Code for Data Slice8.	-

Table 4-130. DM_DESKEW_CON0(Address Offset=0x250)

Field	Bit	R/W	Description	Initial Value
DMDeskew3	[31:24]	R	Write DM De-Skew Code for Data Slice3.	-
DMDeskew2	[23:16]	R	Write DM De-Skew Code for Data Slice2.	-
DMDeskew1	[15:8]	R	Write DM De-Skew Code for Data Slice1.	-

Field	Bit	R/W	Description	Initial Value
DM8Deskew0	[7:0]	R	Write DM De-Skew Code for Data Slice0.	-

Table 4-131. DM_DESKEW_CON1(Address Offset=0x254)

Field	Bit	R/W	Description	Initial Value
DMDeskew7	[31:24]	R	Write DM De-Skew Code for Data Slice7.	-
DMDeskew6	[23:16]	R	Write DM De-Skew Code for Data Slice6.	-
DMDeskew5	[15:8]	R	Write DM De-Skew Code for Data Slice5.	-
DMDeskew4	[7:0]	R	Write DM De-Skew Code for Data Slice4.	-

Table 4-132. DM_DESKEW_CON2(Address Offset=0x258)

Field	Bit	R/W	Description	Initial Value
DMDeskew8	[7:0]	R	Write DM De-Skew Code for Data Slice8.	-

- ~~According to Reg_mode[7:0] and WrDeskew_clear, "ctrl_offsetw*"(=OFFSETW_CON0,1,2) will be written to "Write Deskew Code" register instead of "ctrl_offsetw*" itself.~~
  - ~~Reg_mode[7:0]=4'h04 : DQ0 is written by "ctrl_offsetw*".~~
  - ~~Reg_mode[7:0]=8'h14 : DQ1 is written by "ctrl_offsetw*".~~
  - ~~Reg_mode[7:0]=8'h24 : DQ2 is written by "ctrl_offsetw*".~~
  - ~~Reg_mode[7:0]=8'h34 : DQ3 is written by "ctrl_offsetw*".~~
  - ~~Reg_mode[7:0]=8'h44 : DQ4 is written by "ctrl_offsetw*".~~
  - ~~Reg_mode[7:0]=8'h54 : DQ5 is written by "ctrl_offsetw*".~~
  - ~~Reg_mode[7:0]=8'h64 : DQ6 is written by "ctrl_offsetw*".~~
  - ~~Reg_mode[7:0]=8'h74 : DQ7 is written by "ctrl_offsetw*".~~
  - ~~Reg_mode[7:0]=8'h84 : DM is written by "ctrl_offsetw*".~~
  - ~~Reg_mode[7:0]=8'h94 : All DQ Deskew Codes is cleared.~~

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

4.19 Valid Window Margin Register (All DQ bits)

Table 4-133. VWMC_STAT0(Address Offset=0x25C)

Field	Bit	R/W	Description	Initial Value
D0_VWMC3	[31:24]	R	DQ0 Centering Code for Data Slice3.	-
D0_VWMC2	[23:16]	R	DQ0 Centering Code for Data Slice2.	-
D0_VWMC1	[15:8]	R	DQ0 Centering Code for Data Slice1.	-

Field	Bit	R/W	Description	Initial Value
D0_VWMC0	[7:0]	R	DQ0 Centering Code for Data Slice0.	-

Table 4-134. VWMC_STAT1(Address Offset=0x260)

Field	Bit	R/W	Description	Initial Value
D0_VWMC7	[31:24]	R	DQ0 Centering Code for Data Slice7.	-
D0_VWMC6	[23:16]	R	DQ0 Centering Code for Data Slice6.	-
D0_VWMC5	[15:8]	R	DQ0 Centering Code for Data Slice5.	-
D0_VWMC4	[7:0]	R	DQ0 Centering Code for Data Slice4.	-

Table 4-135. VWMC_STAT2(Address Offset=0x264)

Field	Bit	R/W	Description	Initial Value
D0_VWMC8	[7:0]	R	DQ0 Centering Code for Data Slice8.	-

Table 4-136. VWMC_STAT3(Address Offset=0x268)

Field	Bit	R/W	Description	Initial Value
D1_VWMC3	[31:24]	R	DQ1 Centering Code for Data Slice3.	-
D1_VWMC2	[23:16]	R	DQ1 Centering Code for Data Slice2.	-
D1_VWMC1	[15:8]	R	DQ1 Centering Code for Data Slice1.	-
D1_VWMC0	[7:0]	R	DQ1 Centering Code for Data Slice0.	-

Table 4-137. VWMC_STAT4(Address Offset=0x26C)

Field	Bit	R/W	Description	Initial Value
D1_VWMC7	[31:24]	R	DQ1 Centering Code for Data Slice7.	-
D1_VWMC6	[23:16]	R	DQ1 Centering Code for Data Slice6.	-
D1_VWMC5	[15:8]	R	DQ1 Centering Code for Data Slice5.	-
D1_VWMC4	[7:0]	R	DQ1 Centering Code for Data Slice4.	-

Table 4-138. VWMC_STAT5(Address Offset=0x270)

Field	Bit	R/W	Description	Initial Value
D1_VWMC8	[7:0]	R	DQ1 Centering Code for Data Slice8.	-

Table 4-139. VWMC_STAT6(Address Offset=0x274)

Field	Bit	R/W	Description	Initial Value
D2_VWMC3	[31:24]	R	DQ2 Centering Code for Data Slice3.	-
D2_VWMC2	[23:16]	R	DQ2 Centering Code for Data Slice2.	-
D2_VWMC1	[15:8]	R	DQ2 Centering Code for Data Slice1.	-
D2_VWMC0	[7:0]	R	DQ2 Centering Code for Data Slice0.	-

Table 4-140. VWMC_STAT7(Address Offset=0x278)

Field	Bit	R/W	Description	Initial Value
D2_VWMC7	[31:24]	R	DQ2 Centering Code for Data Slice7.	-
D2_VWMC6	[23:16]	R	DQ2 Centering Code for Data Slice6.	-
D2_VWMC5	[15:8]	R	DQ2 Centering Code for Data Slice5.	-
D2_VWMC4	[7:0]	R	DQ2 Centering Code for Data Slice4.	-

Table 4-141. VWMC_STAT8(Address Offset=0x27C)

Field	Bit	R/W	Description	Initial Value
D2_VWMC8	[7:0]	R	DQ2 Centering Code for Data Slice8.	-

Table 4-142. VWMC_STAT9(Address Offset=0x280)

Field	Bit	R/W	Description	Initial Value
D3_VWMC3	[31:24]	R	DQ3 Centering Code for Data Slice3.	-
D3_VWMC2	[23:16]	R	DQ3 Centering Code for Data Slice2.	-
D3_VWMC1	[15:8]	R	DQ3 Centering Code for Data Slice1.	-
D3_VWMC0	[7:0]	R	DQ3 Centering Code for Data Slice0.	-

Table 4-143. VWMC_STAT10(Address Offset=0x284)

Field	Bit	R/W	Description	Initial Value
D3_VWMC7	[31:24]	R	DQ3 Centering Code for Data Slice7.	-
D3_VWMC6	[23:16]	R	DQ3 Centering Code for Data Slice6.	-
D3_VWMC5	[15:8]	R	DQ3 Centering Code for Data Slice5.	-
D3_VWMC4	[7:0]	R	DQ3 Centering Code for Data Slice4.	-

Table 4-144. VWMC_STAT11(Address Offset=0x288)

Field	Bit	R/W	Description	Initial Value
D3_VWMC8	[7:0]	R	DQ3 Centering Code for Data Slice8.	-

Table 4-145. VWMC_STAT12(Address Offset=0x28C)

Field	Bit	R/W	Description	Initial Value
D4_VWMC3	[31:24]	R	DQ4 Centering Code for Data Slice3.	-
D4_VWMC2	[23:16]	R	DQ4 Centering Code for Data Slice2.	-
D4_VWMC1	[15:8]	R	DQ4 Centering Code for Data Slice1.	-
D4_VWMC0	[7:0]	R	DQ4 Centering Code for Data Slice0.	-

Table 4-146. VWMC_STAT13(Address Offset=0x290)

Field	Bit	R/W	Description	Initial Value
D4_VWMC7	[31:24]	R	DQ4 Centering Code for Data Slice7.	-
D4_VWMC6	[23:16]	R	DQ4 Centering Code for Data Slice6.	-
D4_VWMC5	[15:8]	R	DQ4 Centering Code for Data Slice5.	-
D4_VWMC4	[7:0]	R	DQ4 Centering Code for Data Slice4.	-

Table 4-147. VWMC_STAT14(Address Offset=0x294)

Field	Bit	R/W	Description	Initial Value
D4_VWMC8	[7:0]	R	DQ4 Centering Code for Data Slice8.	-

Table 4-148. VWMC_STAT15(Address Offset=0x298)

Field	Bit	R/W	Description	Initial Value
D5_VWMC3	[31:24]	R	DQ5 Centering Code for Data Slice3.	-
D5_VWMC2	[23:16]	R	DQ5 Centering Code for Data Slice2.	-
D5_VWMC1	[15:8]	R	DQ5 Centering Code for Data Slice1.	-
D5_VWMC0	[7:0]	R	DQ5 Centering Code for Data Slice0.	-

Table 4-149. VWMC_STAT16(Address Offset=0x29C)

Field	Bit	R/W	Description	Initial Value
D5_VWMC7	[31:24]	R	DQ5 Centering Code for Data Slice7.	-
D5_VWMC6	[23:16]	R	DQ5 Centering Code for Data Slice6.	-

Field	Bit	R/W	Description	Initial Value
D5_VWMC5	[15:8]	R	DQ5 Centering Code for Data Slice5.	-
D5_VWMC4	[7:0]	R	DQ5 Centering Code for Data Slice4.	-

Table 4-150. VWMC_STAT17(Address Offset=0x2A0)

Field	Bit	R/W	Description	Initial Value
D5_VWMC8	[7:0]	R	DQ5 Centering Code for Data Slice8.	-

Table 4-151. VWMC_STAT18(Address Offset=0x2A4)

Field	Bit	R/W	Description	Initial Value
D6_VWMC3	[31:24]	R	DQ6 Centering Code for Data Slice3.	-
D6_VWMC2	[23:16]	R	DQ6 Centering Code for Data Slice2.	-
D6_VWMC1	[15:8]	R	DQ6 Centering Code for Data Slice1.	-
D6_VWMC0	[7:0]	R	DQ6 Centering Code for Data Slice0.	-

Table 4-152. VWMC_STAT19(Address Offset=0x2A8)

Field	Bit	R/W	Description	Initial Value
D6_VWMC7	[31:24]	R	DQ6 Centering Code for Data Slice7.	-
D6_VWMC6	[23:16]	R	DQ6 Centering Code for Data Slice6.	-
D6_VWMC5	[15:8]	R	DQ6 Centering Code for Data Slice5.	-
D6_VWMC4	[7:0]	R	DQ6 Centering Code for Data Slice4.	-

Table 4-153. VWMC_STAT20(Address Offset=0x2AC)

Field	Bit	R/W	Description	Initial Value
D6_VWMC8	[7:0]	R	DQ6 Centering Code for Data Slice8.	-

Table 4-154. VWMC_STAT21(Address Offset=0x2B0)

Field	Bit	R/W	Description	Initial Value
D7_VWMC3	[31:24]	R	DQ7 Centering Code for Data Slice3.	-
D7_VWMC2	[23:16]	R	DQ7 Centering Code for Data Slice2.	-
D7_VWMC1	[15:8]	R	DQ7 Centering Code for Data Slice1.	-
D7_VWMC0	[7:0]	R	DQ7 Centering Code for Data Slice0.	-

Table 4-155. VWMC_STAT22(Address Offset=0x2B4)

Field	Bit	R/W	Description	Initial Value
D7_VWMC7	[31:24]	R	DQ7 Centering Code for Data Slice7.	-
D7_VWMC6	[23:16]	R	DQ7 Centering Code for Data Slice6.	-
D7_VWMC5	[15:8]	R	DQ7 Centering Code for Data Slice5.	-
D7_VWMC4	[7:0]	R	DQ7 Centering Code for Data Slice4.	-

Table 4-156. VWMC_STAT23(Address Offset=0x2B8)

Field	Bit	R/W	Description	Initial Value
D7_VWMC8	[7:0]	R	DQ7 Centering Code for Data Slice8.	-

Table 4-157. DM_VWMC_STAT0(Address Offset=0x2BC)

Field	Bit	R/W	Description	Initial Value
DM_VWMC3	[31:24]	R	DM Centering Code Code for Data Slice3.	-
DM_VWMC2	[23:16]	R	DM Centering Code Code for Data Slice2.	-
DM_VWMC1	[15:8]	R	DM Centering Code Code for Data Slice1.	-
DM_VWMC0	[7:0]	R	DM Centering Code Code for Data Slice0.	-

Table 4-158. DM_VWMC_STAT1(Address Offset=0x2C0)

Field	Bit	R/W	Description	Initial Value
DM_VWMC7	[31:24]	R	DM Centering Code for Data Slice7.	-
DM_VWMC6	[23:16]	R	DM Centering Code for Data Slice6.	-
DM_VWMC5	[15:8]	R	DM Centering Code for Data Slice5.	-
DM_VWMC4	[7:0]	R	DM Centering Code for Data Slice4.	-

Table 4-159. DM_VWMC_STAT2(Address Offset=0x2C4)

Field	Bit	R/W	Description	Initial Value
DM_VWMC8	[7:0]	R	DM Centering Code for Data Slice8.	-

Table 4-160. VWML_STAT0(Address Offset=0x2C8)

Field	Bit	R/W	Description	Initial Value
D0_VWML3	[31:24]	R	DQ0 Left Code for Data Slice3.	-
D0_VWML2	[23:16]	R	DQ0 Left Code for Data Slice2.	-
D0_VWML1	[15:8]	R	DQ0 Left Code for Data Slice1.	-
D0_VWML0	[7:0]	R	DQ0 Left Code for Data Slice0.	-

Table 4-161. VWML_STAT1(Address Offset=0x2CC)

Field	Bit	R/W	Description	Initial Value
D0_VWML7	[31:24]	R	DQ0 Left Code for Data Slice7.	-
D0_VWML6	[23:16]	R	DQ0 Left Code for Data Slice6.	-
D0_VWML5	[15:8]	R	DQ0 Left Code for Data Slice5.	-
D0_VWML4	[7:0]	R	DQ0 Left Code for Data Slice4.	-

Table 4-162. VWML_STAT2(Address Offset=0x2D0)

Field	Bit	R/W	Description	Initial Value
D0_VWML8	[7:0]	R	DQ0 Left Code for Data Slice8.	-

Table 4-163. VWML_STAT3(Address Offset=0x2D4)

Field	Bit	R/W	Description	Initial Value
D1_VWML3	[31:24]	R	DQ1 Left Code for Data Slice3.	-
D1_VWML2	[23:16]	R	DQ1 Left Code for Data Slice2.	-
D1_VWML1	[15:8]	R	DQ1 Left Code for Data Slice1.	-
D1_VWML0	[7:0]	R	DQ1 Left Code for Data Slice0.	-

Table 4-164. VWML_STAT4(Address Offset=0x2D8)

Field	Bit	R/W	Description	Initial Value
D1_VWML7	[31:24]	R	DQ1 Left Code for Data Slice7.	-
D1_VWML6	[23:16]	R	DQ1 Left Code for Data Slice6.	-
D1_VWML5	[15:8]	R	DQ1 Left Code for Data Slice5.	-
D1_VWML4	[7:0]	R	DQ1 Left Code for Data Slice4.	-

Table 4-165. VWML_STAT5(Address Offset=0x2DC)

Field	Bit	R/W	Description	Initial Value
D1_VWML8	[7:0]	R	DQ1 Left Code for Data Slice8.	-

Table 4-166. VWML_STAT6(Address Offset=0x2E0)

Field	Bit	R/W	Description	Initial Value
D2_VWML3	[31:24]	R	DQ2 Left Code for Data Slice3.	-
D2_VWML2	[23:16]	R	DQ2 Left Code for Data Slice2.	-
D2_VWML1	[15:8]	R	DQ2 Left Code for Data Slice1.	-
D2_VWML0	[7:0]	R	DQ2 Left Code for Data Slice0.	-

Table 4-167. VWML_STAT7(Address Offset=0x2E4)

Field	Bit	R/W	Description	Initial Value
D2_VWML7	[31:24]	R	DQ2 Left Code for Data Slice7.	-
D2_VWML6	[23:16]	R	DQ2 Left Code for Data Slice6.	-
D2_VWML5	[15:8]	R	DQ2 Left Code for Data Slice5.	-
D2_VWML4	[7:0]	R	DQ2 Left Code for Data Slice4.	-

Table 4-168. VWML_STAT8(Address Offset=0x2E8)

Field	Bit	R/W	Description	Initial Value
D2_VWML8	[7:0]	R	DQ2 Left Code for Data Slice8.	-

Table 4-169. VWML_STAT9(Address Offset=0x2EC)

Field	Bit	R/W	Description	Initial Value
D3_VWML3	[31:24]	R	DQ3 Left Code for Data Slice3.	-
D3_VWML2	[23:16]	R	DQ3 Left Code for Data Slice2.	-
D3_VWML1	[15:8]	R	DQ3 Left Code for Data Slice1.	-
D3_VWML0	[7:0]	R	DQ3 Left Code for Data Slice0.	-

Table 4-170. VWML_STAT10(Address Offset=0x2F0)

Field	Bit	R/W	Description	Initial Value
D3_VWML7	[31:24]	R	DQ3 Left Code for Data Slice7.	-
D3_VWML6	[23:16]	R	DQ3 Left Code for Data Slice6.	-

Field	Bit	R/W	Description	Initial Value
D3_VWML5	[15:8]	R	DQ3 Left Code for Data Slice5.	-
D3_VWML4	[7:0]	R	DQ3 Left Code for Data Slice4.	-

Table 4-171. VWML_STAT11(Address Offset=0x2F4)

Field	Bit	R/W	Description	Initial Value
D3_VWML8	[7:0]	R	DQ3 Left Code for Data Slice8.	-

Table 4-172. VWML_STAT12(Address Offset=0x2F8)

Field	Bit	R/W	Description	Initial Value
D4_VWML3	[31:24]	R	DQ4 Left Code for Data Slice3.	-
D4_VWML2	[23:16]	R	DQ4 Left Code for Data Slice2.	-
D4_VWML1	[15:8]	R	DQ4 Left Code for Data Slice1.	-
D4_VWML0	[7:0]	R	DQ4 Left Code for Data Slice0.	-

Table 4-173. VWML_STAT13(Address Offset=0x2FC)

Field	Bit	R/W	Description	Initial Value
D4_VWML7	[31:24]	R	DQ4 Left Code for Data Slice7.	-
D4_VWML6	[23:16]	R	DQ4 Left Code for Data Slice6.	-
D4_VWML5	[15:8]	R	DQ4 Left Code for Data Slice5.	-
D4_VWML4	[7:0]	R	DQ4 Left Code for Data Slice4.	-

Table 4-174. VWML_STAT14(Address Offset=0x300)

Field	Bit	R/W	Description	Initial Value
D4_VWML8	[7:0]	R	DQ4 Left Code for Data Slice8.	-

Table 4-175. VWML_STAT15(Address Offset=0x304)

Field	Bit	R/W	Description	Initial Value
D5_VWML3	[31:24]	R	DQ5 Left Code for Data Slice3.	-
D5_VWML2	[23:16]	R	DQ5 Left Code for Data Slice2.	-
D5_VWML1	[15:8]	R	DQ5 Left Code for Data Slice1.	-
D5_VWML0	[7:0]	R	DQ5 Left Code for Data Slice0.	-

Table 4-176. VWML_STAT16(Address Offset=0x308)

Field	Bit	R/W	Description	Initial Value
D5_VWML7	[31:24]	R	DQ5 Left Code for Data Slice7.	-
D5_VWML6	[23:16]	R	DQ5 Left Code for Data Slice6.	-
D5_VWML5	[15:8]	R	DQ5 Left Code for Data Slice5.	-
D5_VWML4	[7:0]	R	DQ5 Left Code for Data Slice4.	-

Table 4-177. VWML_STAT17(Address Offset=0x30C)

Field	Bit	R/W	Description	Initial Value
D5_VWML8	[7:0]	R	DQ5 Left Code for Data Slice8.	-

Table 4-178. VWML_STAT18(Address Offset=0x310)

Field	Bit	R/W	Description	Initial Value
D6_VWML3	[31:24]	R	DQ6 Left Code for Data Slice3.	-
D6_VWML2	[23:16]	R	DQ6 Left Code for Data Slice2.	-
D6_VWML1	[15:8]	R	DQ6 Left Code for Data Slice1.	-
D6_VWML0	[7:0]	R	DQ6 Left Code for Data Slice0.	-

Table 4-179. VWML_STAT19(Address Offset=0x314)

Field	Bit	R/W	Description	Initial Value
D6_VWML7	[31:24]	R	DQ6 Left Code for Data Slice7.	-
D6_VWML6	[23:16]	R	DQ6 Left Code for Data Slice6.	-
D6_VWML5	[15:8]	R	DQ6 Left Code for Data Slice5.	-
D6_VWML4	[7:0]	R	DQ6 Left Code for Data Slice4.	-

Table 4-180. VWML_STAT20(Address Offset=0x318)

Field	Bit	R/W	Description	Initial Value
D6_VWML8	[7:0]	R	DQ6 Left Code for Data Slice8.	-

Table 4-181. VWML_STAT21(Address Offset=0x31C)

Field	Bit	R/W	Description	Initial Value
D7_VWML3	[31:24]	R	DQ7 Left Code for Data Slice3.	-
D7_VWML2	[23:16]	R	DQ7 Left Code for Data Slice2.	-
D7_VWML1	[15:8]	R	DQ7 Left Code for Data Slice1.	-
D7_VWML0	[7:0]	R	DQ7 Left Code for Data Slice0.	-

Table 4-182. VWML_STAT22(Address Offset=0x320)

Field	Bit	R/W	Description	Initial Value
D7_VWML7	[31:24]	R	DQ7 Left Code for Data Slice7.	-
D7_VWML6	[23:16]	R	DQ7 Left Code for Data Slice6.	-
D7_VWML5	[15:8]	R	DQ7 Left Code for Data Slice5.	-
D7_VWML4	[7:0]	R	DQ7 Left Code for Data Slice4.	-

Table 4-183. VWML_STAT23(Address Offset=0x324)

Field	Bit	R/W	Description	Initial Value
D7_VWML8	[7:0]	R	DQ7 Left Code for Data Slice8.	-

Table 4-184. DM_VWML_STAT0(Address Offset=0x328)

Field	Bit	R/W	Description	Initial Value
DM_VWML3	[31:24]	R	DM Left Code Code for Data Slice3.	-
DM_VWML2	[23:16]	R	DM Left Code Code for Data Slice2.	-
DM_VWML1	[15:8]	R	DM Left Code Code for Data Slice1.	-
DM_VWML0	[7:0]	R	DM Left Code Code for Data Slice0.	-

Table 4-185. DM_VWML_STAT1(Address Offset=0x32C)

Field	Bit	R/W	Description	Initial Value
DM_VWML7	[31:24]	R	DM Left Code for Data Slice7.	-
DM_VWML6	[23:16]	R	DM Left Code for Data Slice6.	-
DM_VWML5	[15:8]	R	DM Left Code for Data Slice5.	-
DM_VWML4	[7:0]	R	DM Left Code for Data Slice4.	-

Table 4-186. DM_VWML_STAT2(Address Offset=0x330)

Field	Bit	R/W	Description	Initial Value
DM_VWML8	[7:0]	R	DM Left Code for Data Slice8.	-

Table 4-187. VWMR_STAT0(Address Offset=0x334)

Field	Bit	R/W	Description	Initial Value
D0_VWMR3	[31:24]	R	DQ0 Right Code for Data Slice3.	-
D0_VWMR2	[23:16]	R	DQ0 Right Code for Data Slice2.	-
D0_VWMR1	[15:8]	R	DQ0 Right Code for Data Slice1.	-
D0_VWMR0	[7:0]	R	DQ0 Right Code for Data Slice0.	-

Table 4-188. VWMR_STAT1(Address Offset=0x338)

Field	Bit	R/W	Description	Initial Value
D0_VWMR7	[31:24]	R	DQ0 Right Code for Data Slice7.	-
D0_VWMR6	[23:16]	R	DQ0 Right Code for Data Slice6.	-
D0_VWMR5	[15:8]	R	DQ0 Right Code for Data Slice5.	-
D0_VWMR4	[7:0]	R	DQ0 Right Code for Data Slice4.	-

Table 4-189. VWMR_STAT2(Address Offset=0x33C)

Field	Bit	R/W	Description	Initial Value
D0_VWMR8	[7:0]	R	DQ0 Right Code for Data Slice8.	-

Table 4-190. VWMR_STAT3(Address Offset=0x340)

Field	Bit	R/W	Description	Initial Value
D1_VWMR3	[31:24]	R	DQ1 Right Code for Data Slice3.	-
D1_VWMR2	[23:16]	R	DQ1 Right Code for Data Slice2.	-
D1_VWMR1	[15:8]	R	DQ1 Right Code for Data Slice1.	-
D1_VWMR0	[7:0]	R	DQ1 Right Code for Data Slice0.	-

Table 4-191. VWMR_STAT4(Address Offset=0x344)

Field	Bit	R/W	Description	Initial Value
D1_VWMR7	[31:24]	R	DQ1 Right Code for Data Slice7.	-
D1_VWMR6	[23:16]	R	DQ1 Right Code for Data Slice6.	-

Field	Bit	R/W	Description	Initial Value
D1_VWMR5	[15:8]	R	DQ1 Right Code for Data Slice5.	-
D1_VWMR4	[7:0]	R	DQ1 Right Code for Data Slice4.	-

Table 4-192. VWMR_STAT5(Address Offset=0x348)

Field	Bit	R/W	Description	Initial Value
D1_VWMR8	[7:0]	R	DQ1 Right Code for Data Slice8.	-

Table 4-193. VWMR_STAT6(Address Offset=0x34C)

Field	Bit	R/W	Description	Initial Value
D2_VWMR3	[31:24]	R	DQ2 Right Code for Data Slice3.	-
D2_VWMR2	[23:16]	R	DQ2 Right Code for Data Slice2.	-
D2_VWMR1	[15:8]	R	DQ2 Right Code for Data Slice1.	-
D2_VWMR0	[7:0]	R	DQ2 Right Code for Data Slice0.	-

Table 4-194. VWMR_STAT7(Address Offset=0x350)

Field	Bit	R/W	Description	Initial Value
D2_VWMR7	[31:24]	R	DQ2 Right Code for Data Slice7.	-
D2_VWMR6	[23:16]	R	DQ2 Right Code for Data Slice6.	-
D2_VWMR5	[15:8]	R	DQ2 Right Code for Data Slice5.	-
D2_VWMR4	[7:0]	R	DQ2 Right Code for Data Slice4.	-

Table 4-195. VWMR_STAT8(Address Offset=0x354)

Field	Bit	R/W	Description	Initial Value
D2_VWMR8	[7:0]	R	DQ2 Right Code for Data Slice8.	-

Table 4-196. VWMR_STAT9(Address Offset=0x358)

Field	Bit	R/W	Description	Initial Value
D3_VWMR3	[31:24]	R	DQ3 Right Code for Data Slice3.	-
D3_VWMR2	[23:16]	R	DQ3 Right Code for Data Slice2.	-
D3_VWMR1	[15:8]	R	DQ3 Right Code for Data Slice1.	-
D3_VWMR0	[7:0]	R	DQ3 Right Code for Data Slice0.	-

Table 4-197. VWMR_STAT10(Address Offset=0x35C)

Field	Bit	R/W	Description	Initial Value
D3_VWMR7	[31:24]	R	DQ3 Right Code for Data Slice7.	-
D3_VWMR6	[23:16]	R	DQ3 Right Code for Data Slice6.	-
D3_VWMR5	[15:8]	R	DQ3 Right Code for Data Slice5.	-
D3_VWMR4	[7:0]	R	DQ3 Right Code for Data Slice4.	-

Table 4-198. VWMR_STAT11(Address Offset=0x360)

Field	Bit	R/W	Description	Initial Value
D3_VWMR8	[7:0]	R	DQ3 Right Code for Data Slice8.	-

Table 4-199. VWMR_STAT12(Address Offset=0x364)

Field	Bit	R/W	Description	Initial Value
D4_VWMR3	[31:24]	R	DQ4 Right Code for Data Slice3.	-
D4_VWMR2	[23:16]	R	DQ4 Right Code for Data Slice2.	-
D4_VWMR1	[15:8]	R	DQ4 Right Code for Data Slice1.	-
D4_VWMR0	[7:0]	R	DQ4 Right Code for Data Slice0.	-

Table 4-200. VWMR_STAT13(Address Offset=0x368)

Field	Bit	R/W	Description	Initial Value
D4_VWMR7	[31:24]	R	DQ4 Right Code for Data Slice7.	-
D4_VWMR6	[23:16]	R	DQ4 Right Code for Data Slice6.	-
D4_VWMR5	[15:8]	R	DQ4 Right Code for Data Slice5.	-
D4_VWMR4	[7:0]	R	DQ4 Right Code for Data Slice4.	-

Table 4-201. VWMR_STAT14(Address Offset=0x36C)

Field	Bit	R/W	Description	Initial Value
D4_VWMR8	[7:0]	R	DQ4 Right Code for Data Slice8.	-

Table 4-202. VWMR_STAT15(Address Offset=0x370)

Field	Bit	R/W	Description	Initial Value
D5_VWMR3	[31:24]	R	DQ5 Right Code for Data Slice3.	-
D5_VWMR2	[23:16]	R	DQ5 Right Code for Data Slice2.	-
D5_VWMR1	[15:8]	R	DQ5 Right Code for Data Slice1.	-
D5_VWMR0	[7:0]	R	DQ5 Right Code for Data Slice0.	-

Table 4-203. VWMR_STAT16(Address Offset=0x374)

Field	Bit	R/W	Description	Initial Value
D5_VWMR7	[31:24]	R	DQ5 Right Code for Data Slice7.	-
D5_VWMR6	[23:16]	R	DQ5 Right Code for Data Slice6.	-
D5_VWMR5	[15:8]	R	DQ5 Right Code for Data Slice5.	-
D5_VWMR4	[7:0]	R	DQ5 Right Code for Data Slice4.	-

Table 4-204. VWMR_STAT17(Address Offset=0x378)

Field	Bit	R/W	Description	Initial Value
D5_VWMR8	[7:0]	R	DQ5 Right Code for Data Slice8.	-

Table 4-205. VWMR_STAT18(Address Offset=0x37C)

Field	Bit	R/W	Description	Initial Value
D6_VWMR3	[31:24]	R	DQ6 Right Code for Data Slice3.	-
D6_VWMR2	[23:16]	R	DQ6 Right Code for Data Slice2.	-
D6_VWMR1	[15:8]	R	DQ6 Right Code for Data Slice1.	-
D6_VWMR0	[7:0]	R	DQ6 Right Code for Data Slice0.	-

Table 4-206. VWMR_STAT19(Address Offset=0x380)

Field	Bit	R/W	Description	Initial Value
D6_VWMR7	[31:24]	R	DQ6 Right Code for Data Slice7.	-
D6_VWMR6	[23:16]	R	DQ6 Right Code for Data Slice6.	-
D6_VWMR5	[15:8]	R	DQ6 Right Code for Data Slice5.	-
D6_VWMR4	[7:0]	R	DQ6 Right Code for Data Slice4.	-

Table 4-207. VWMR_STAT20(Address Offset=0x384)

Field	Bit	R/W	Description	Initial Value
D6_VWMR8	[7:0]	R	DQ6 Right Code for Data Slice8.	-

Table 4-208. VWMR_STAT21(Address Offset=0x388)

Field	Bit	R/W	Description	Initial Value
D7_VWMR3	[31:24]	R	DQ7 Right Code for Data Slice3.	-
D7_VWMR2	[23:16]	R	DQ7 Right Code for Data Slice2.	-
D7_VWMR1	[15:8]	R	DQ7 Right Code for Data Slice1.	-
D7_VWMR0	[7:0]	R	DQ7 Right Code for Data Slice0.	-

Table 4-209. VWMR_STAT22(Address Offset=0x38C)

Field	Bit	R/W	Description	Initial Value
D7_VWMR7	[31:24]	R	DQ7 Right Code for Data Slice7.	-
D7_VWMR6	[23:16]	R	DQ7 Right Code for Data Slice6.	-
D7_VWMR5	[15:8]	R	DQ7 Right Code for Data Slice5.	-
D7_VWMR4	[7:0]	R	DQ7 Right Code for Data Slice4.	-

Table 4-210. VWMR_STAT23(Address Offset=0x390)

Field	Bit	R/W	Description	Initial Value
D7_VWMR8	[7:0]	R	DQ7 Right Code for Data Slice8.	-

Table 4-211. DM_VWMR_STAT0(Address Offset=0x394)

Field	Bit	R/W	Description	Initial Value
DM_VWMR3	[31:24]	R	DM Right Code Code for Data Slice3.	-
DM_VWMR2	[23:16]	R	DM Right Code Code for Data Slice2.	-
DM_VWMR1	[15:8]	R	DM Right Code Code for Data Slice1.	-
DM_VWMR0	[7:0]	R	DM Right Code Code for Data Slice0.	-

Table 4-212. DM_VWMR_STAT1(Address Offset=0x398)

Field	Bit	R/W	Description	Initial Value
DM_VWMR7	[31:24]	R	DM Right Code for Data Slice7.	-
DM_VWMR6	[23:16]	R	DM Right Code for Data Slice6.	-

Field	Bit	R/W	Description	Initial Value
DM_VWMR5	[15:8]	R	DM Right Code for Data Slice5.	-
DM_VWMR4	[7:0]	R	DM Right Code for Data Slice4.	-

Table 4-213. DM_VWMR_STAT2(Address Offset=0x39C)

Field	Bit	R/W	Description	Initial Value
DM_VWMR8	[7:0]	R	DM Right Code for Data Slice8.	-

4.20 DQ Status Register

Table 4-214. DQ_IO_RDATA0(Address Offset=0x3A0)

Field	Bit	R/W	Description	Initial Value
DQ_IO_RD3	[31:24]	R	DQ I/O Read Data for DS3	-
DQ_IO_RD2	[23:16]	R	DQ I/O Read Data for DS2	-
DQ_IO_RD1	[15:8]	R	DQ I/O Read Data for DS1	-
DQ_IO_RD0	[7:0]	R	DQ I/O Read Data for DS0	-

Table 4-215. DQ_IO_RDATA1(Address Offset=0x3A4)

Field	Bit	R/W	Description	Initial Value
DQ_IO_RD7	[31:24]	R	DQ I/O Read Data for DS7	-
DQ_IO_RD6	[23:16]	R	DQ I/O Read Data for DS6	-
DQ_IO_RD5	[15:8]	R	DQ I/O Read Data for DS5	-
DQ_IO_RD4	[7:0]	R	DQ I/O Read Data for DS4	-

Table 4-216. DQ_IO_RDATA2(Address Offset=0x3A8)

Field	Bit	R/W	Description	Initial Value
DQ_IO_RD8	[7:0]	R	DQ I/O Read Data for DS8	-

4.21 Version Information Register

Table 4-217. VERSION_INFO_STAT0(Address Offset=0x3AC)

Field	Bit	R/W	Description	Initial Value
Version_Info	[31:0]	R	Version Information	0x0600_0000

4.22 Feedback Control Register

Table 4-218. FB_CON0(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

5 Functional Description

5.1 DLL

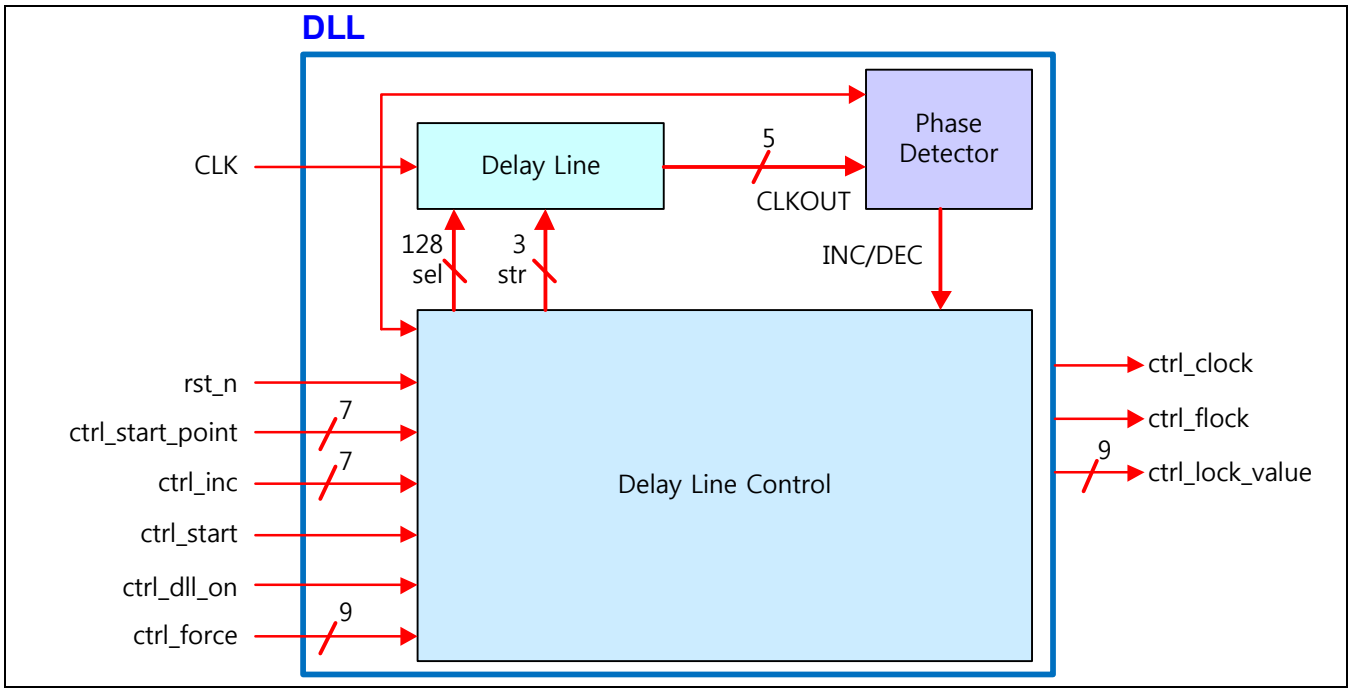

Figure 5-1. DLL Interface

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.

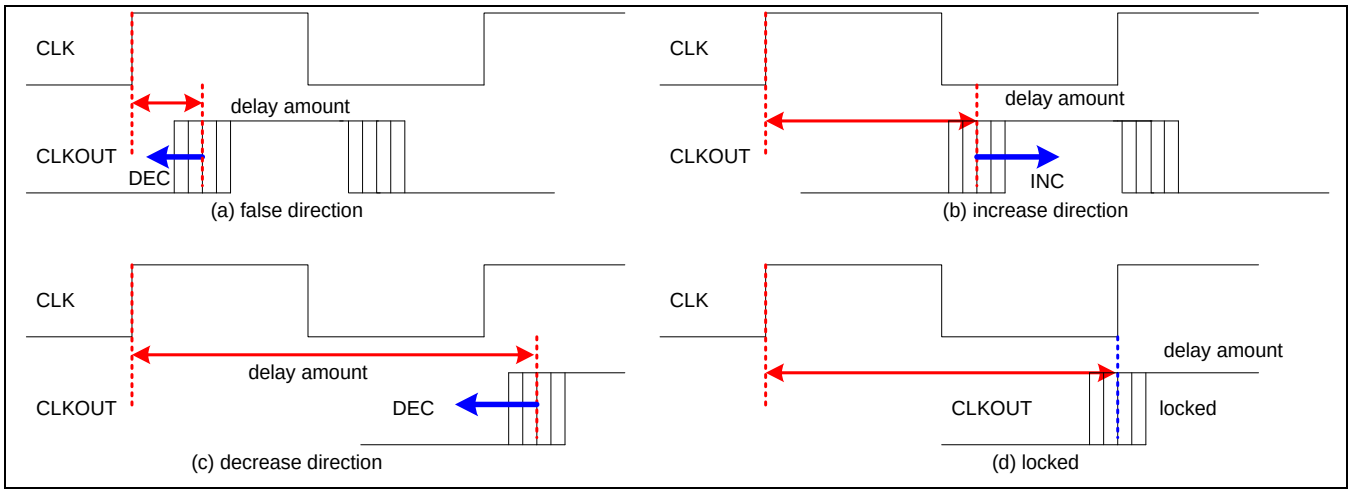

Figure 5-2. DLL lead/lag/lock Cases

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.

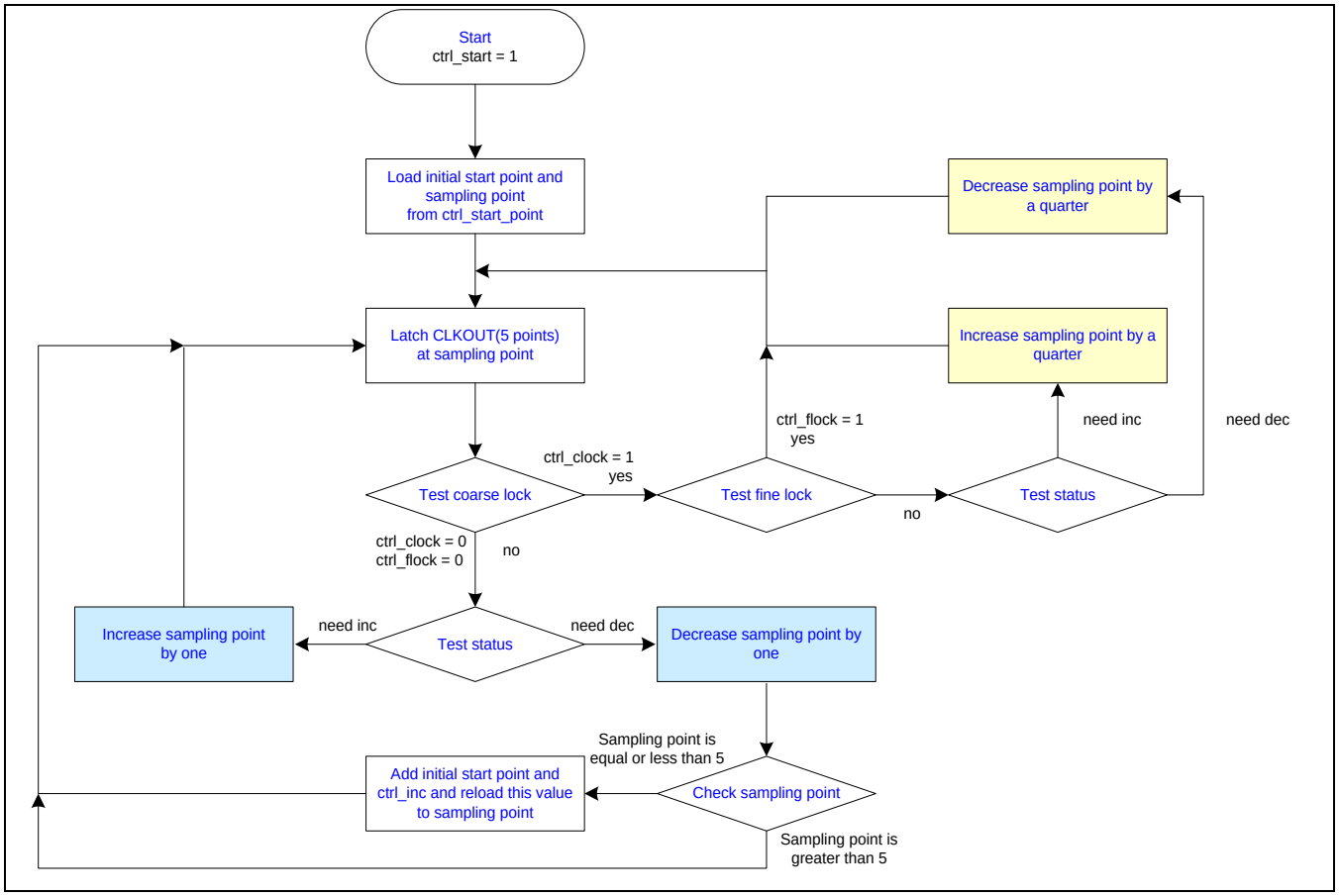

Figure 5-3. DLL Lock Flow Chart

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 RESET

All clocks should be toggled during reset period.

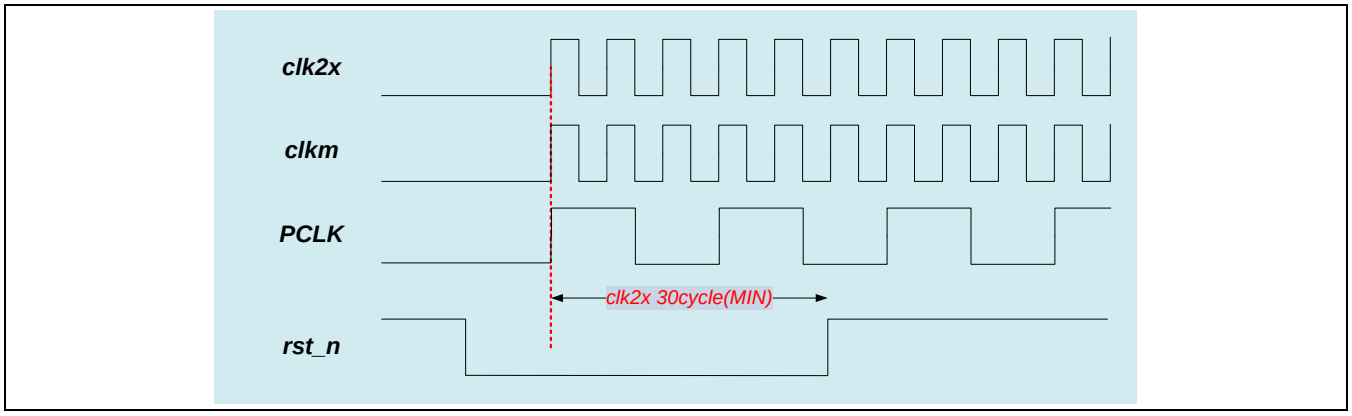

Figure 5-4. The timing requirement between "rst_n" and clock

5.3 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). The jitter and duty requirement of clkm is the same as that of clk2x. 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.

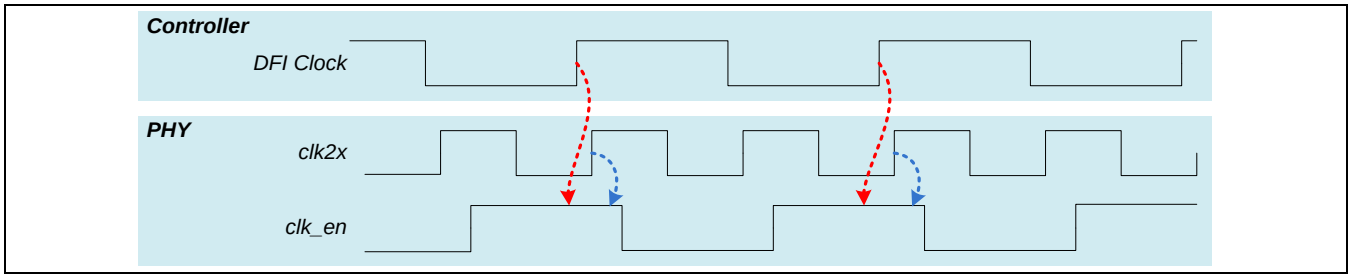

Figure 5-5. Timing Diagram between Controller Clock and PHY Clock(1:2 frequency ratio)

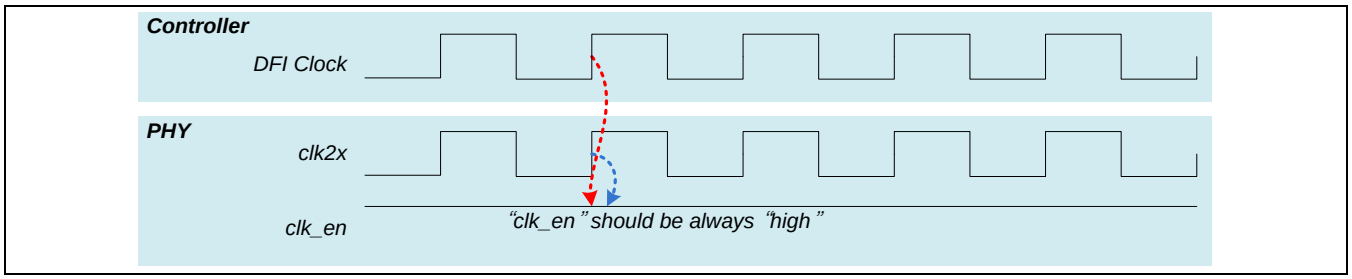

Figure 5-6. Timing Diagram between Controller Clock and PHY Clock(1:1 frequency ratio)

5.4 CONTROL PATH

Signal interface timing for the 1:2 frequency ratio control path is shown in Figure 5-7. 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.

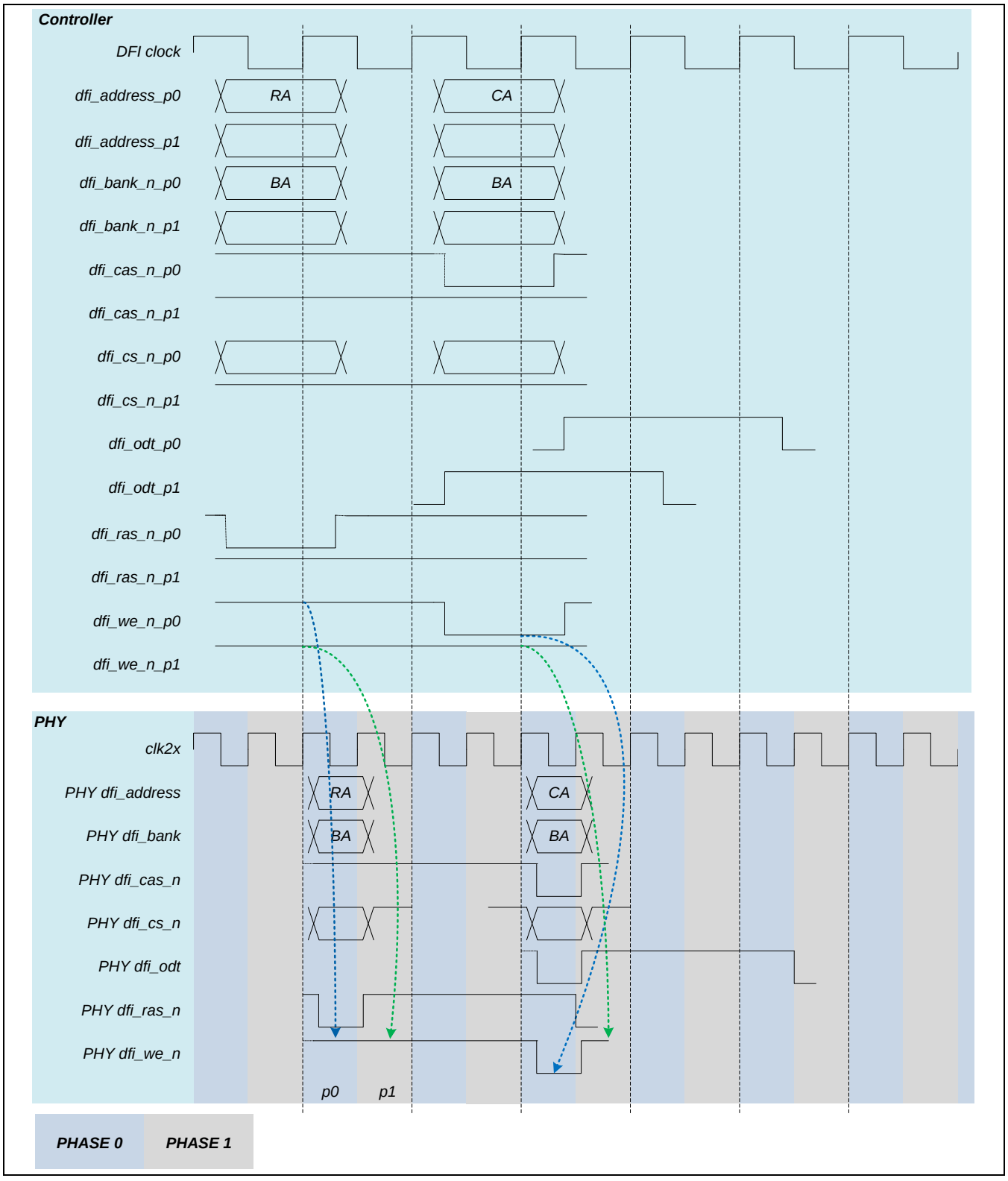

Figure 5-7. Timing Diagram for Frequency Ratio Control Path

5.5 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 5-8*. dfi_wrdata_en_p0/p1 is enable signal to indicate dfi_wrdata_p0/p1 and dfi_mask_p0/p1 are valid.

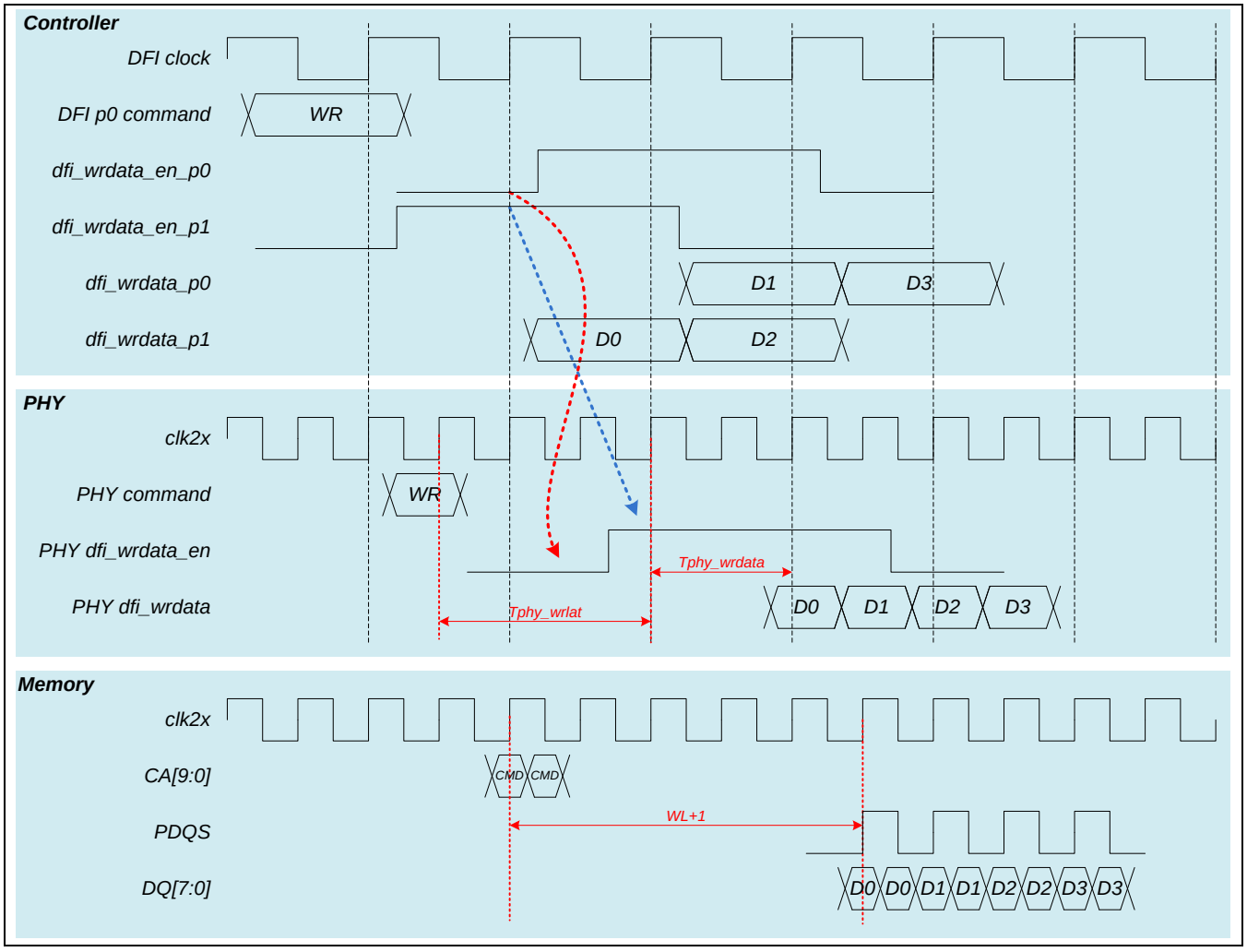

Figure 5-8. Data write Timing Diagram with phase 0 command(1:2 frequency ratio, LPDDR3)

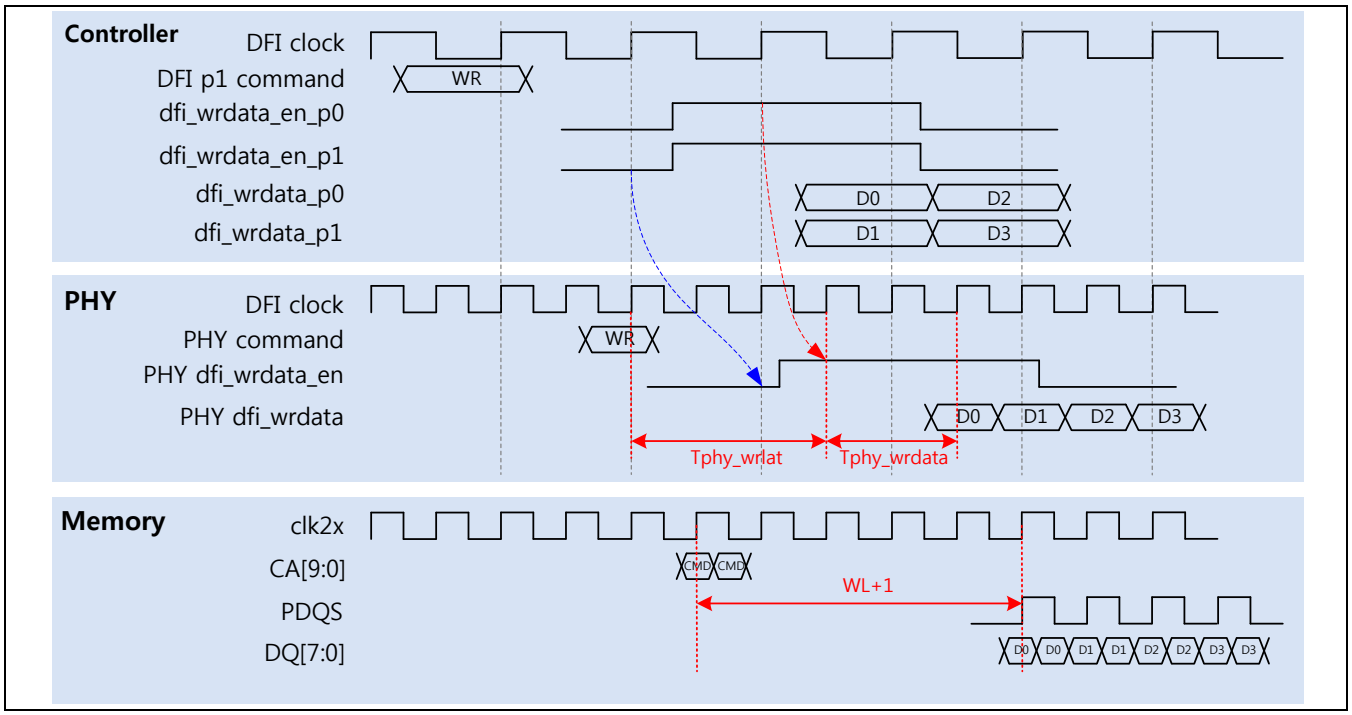

Figure 5-9. Data write Timing Diagram with phase 1 command(1:2 frequency ratio, LPDDR3)

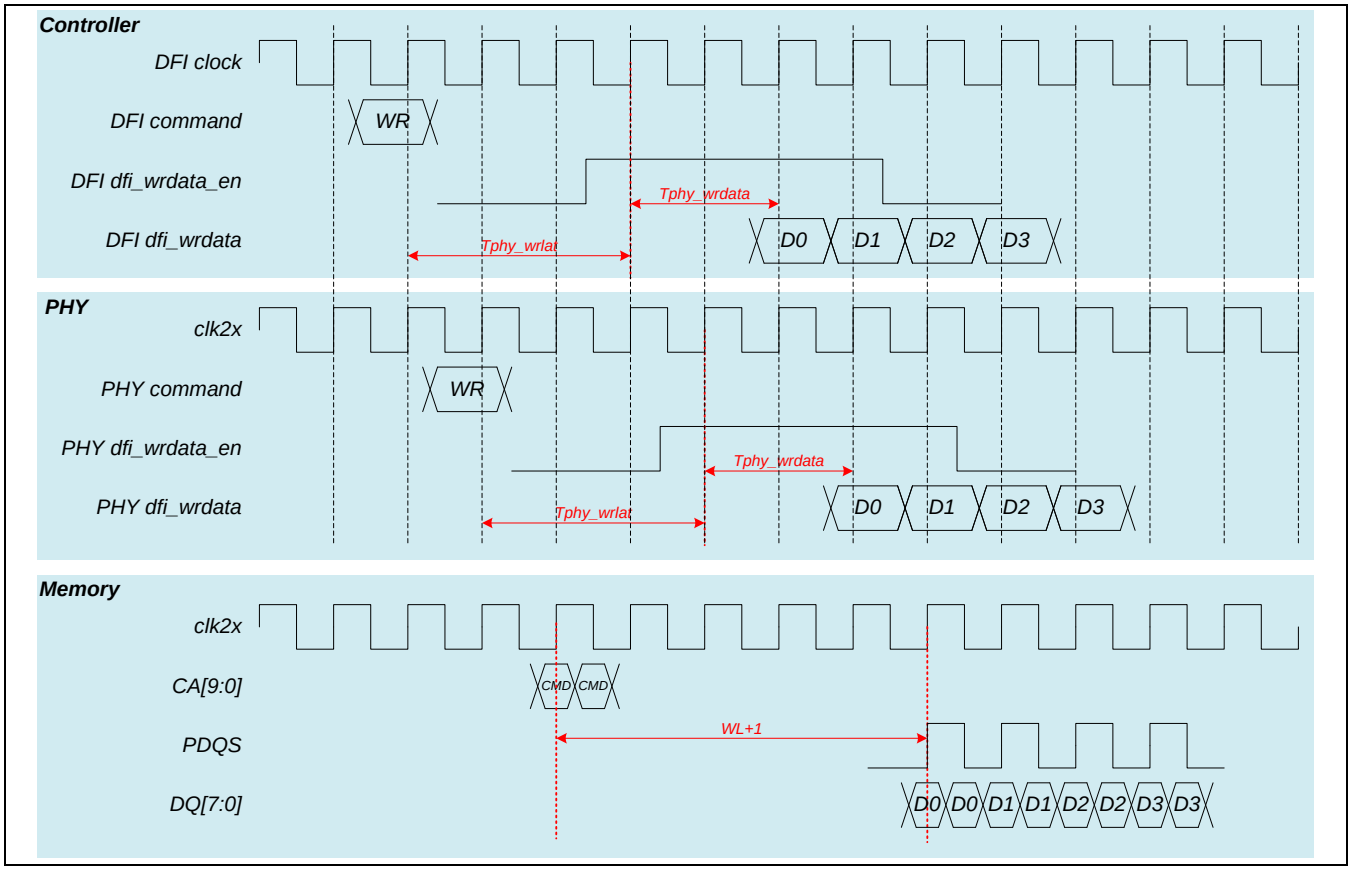

Figure 5-10. Data write Timing Diagram(1:1 frequency ratio, LPDDR3)

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=1+"The maximum value of T0_rddata_en ~ T8_rddata_en in T_RDDATA_CON0, T_RDDATA_CON1, T_RDDATA_CON2"

Signal interface timing for data read is shown in *Figure 5-11*. 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.

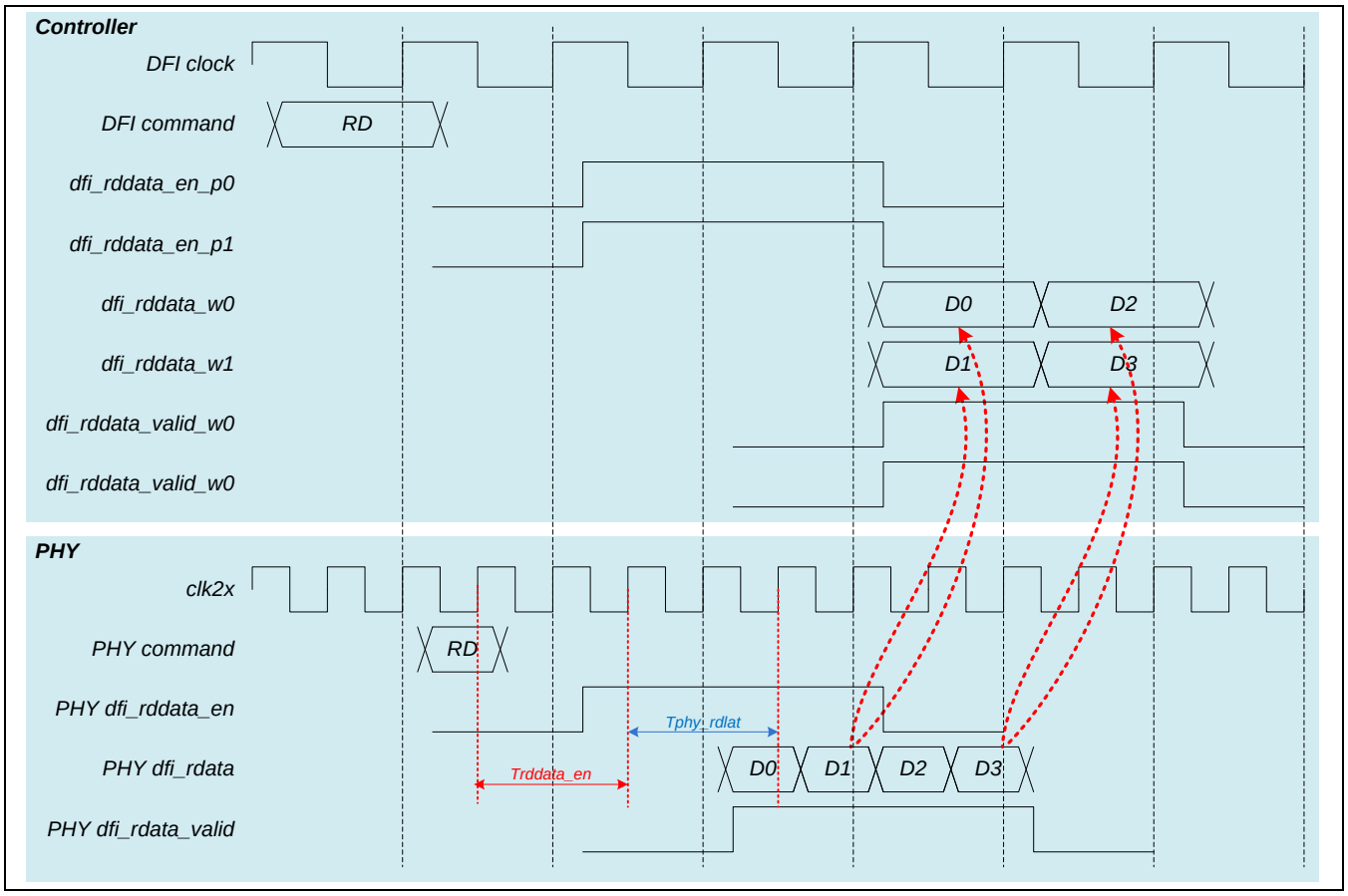

Figure 5-11. Data Read Timing Diagram(1:2 frequency ratio)

Figure 5-12. 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.

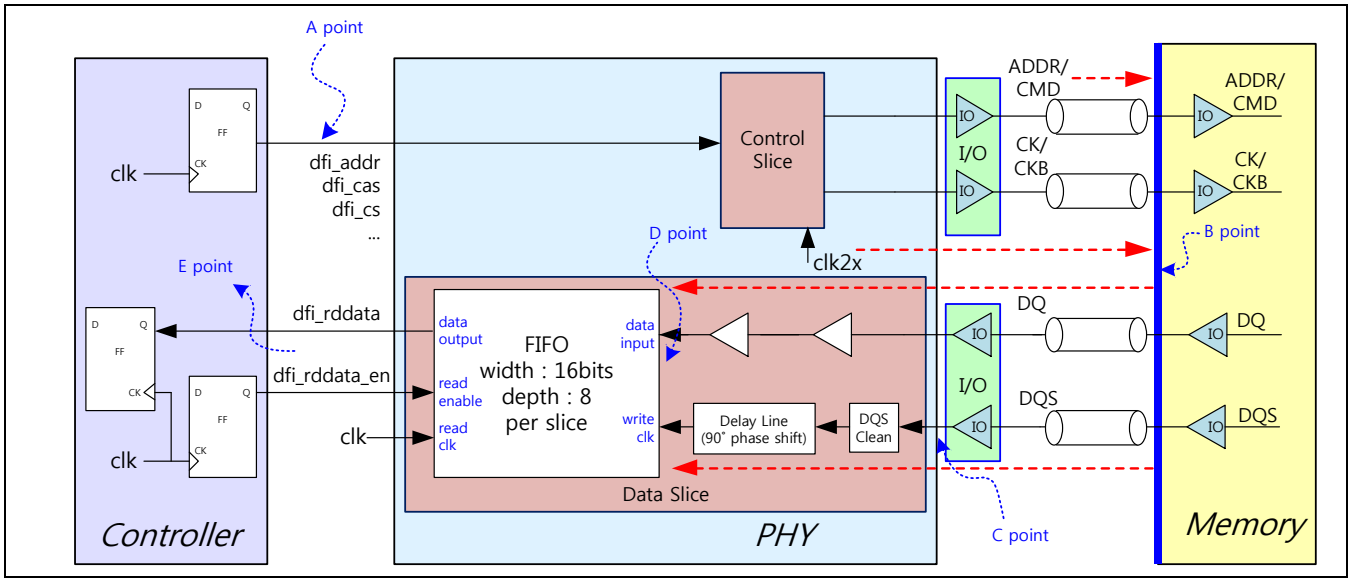

Figure 5-12. Read Data Path

Figure 5-13. 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 5-13. 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 T_RDDATA_CON0). To consider Voltage/ Temperature variations during operation, the maximum value of "Trddata_en" which has got from T_RDDATA_CON0 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.

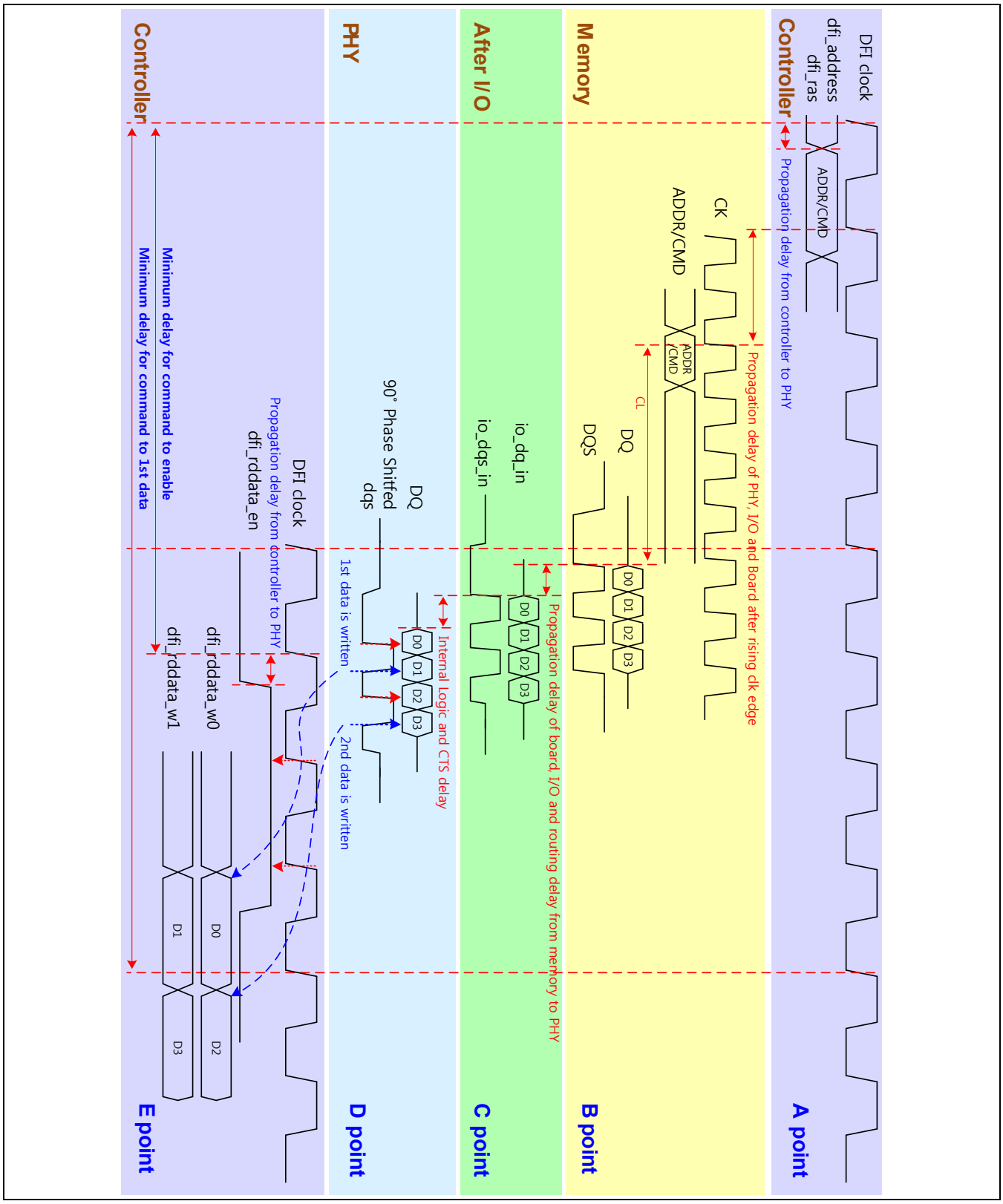

Figure 5-13. Full Read Path Timing Diagram

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(LP_CON0[NS-1:0]) 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.

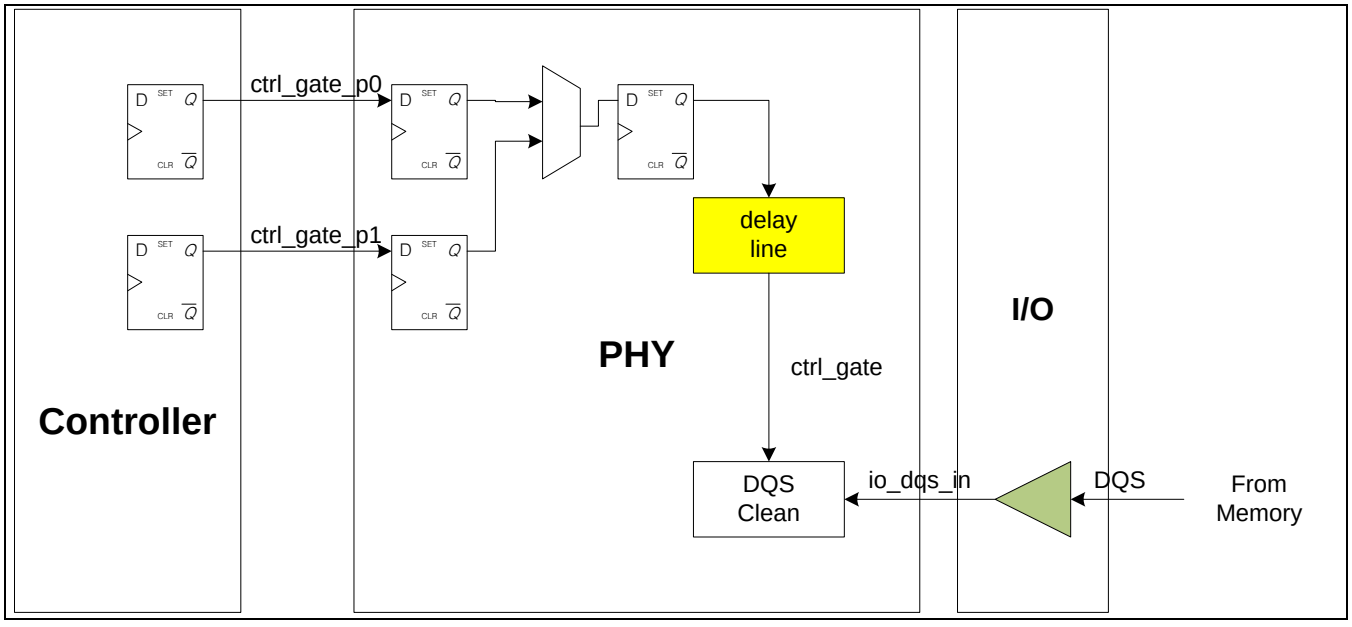

Figure 6-1. Block Diagram of GATE Signal Path

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.

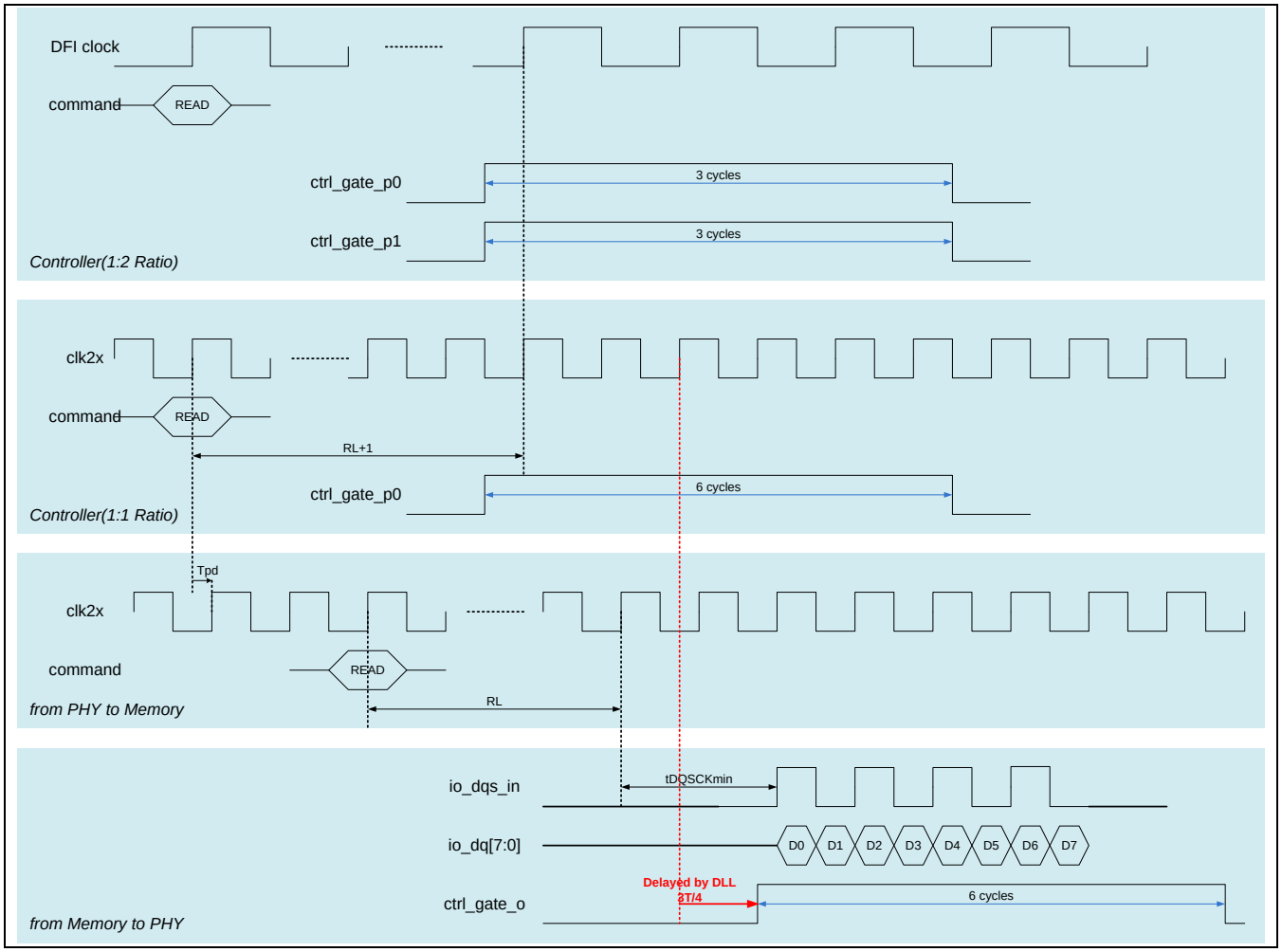

Figure 6-2. GATE Signal when tDQSCK=2.5ns(LPDDR3 800MHz)

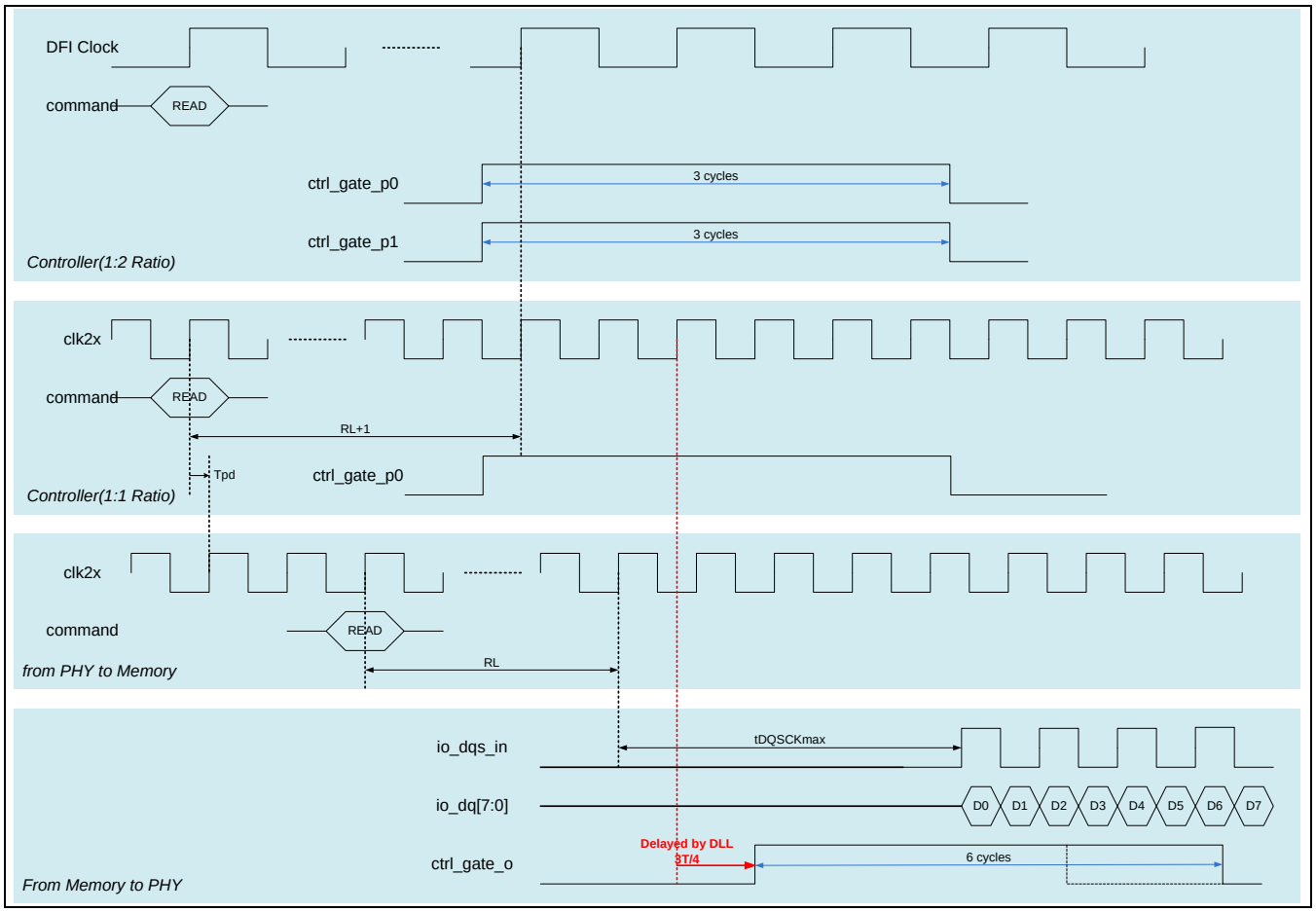

Figure 6-3. GATE Signal when tDQSCK=5.5ns(LPDDR3 800MHz)

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
12	ZQ I/O	ATE ZQ I/O Test Mode	For ATE ZQ I/O Test vector

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
Normal Vector	-	Mandatory
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 Mode	Low-speed	Mandatory
High-Z Test Mode	Low-speed	Mandatory
ZQ I/O Test Mode	Low-speed	Mandatory

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

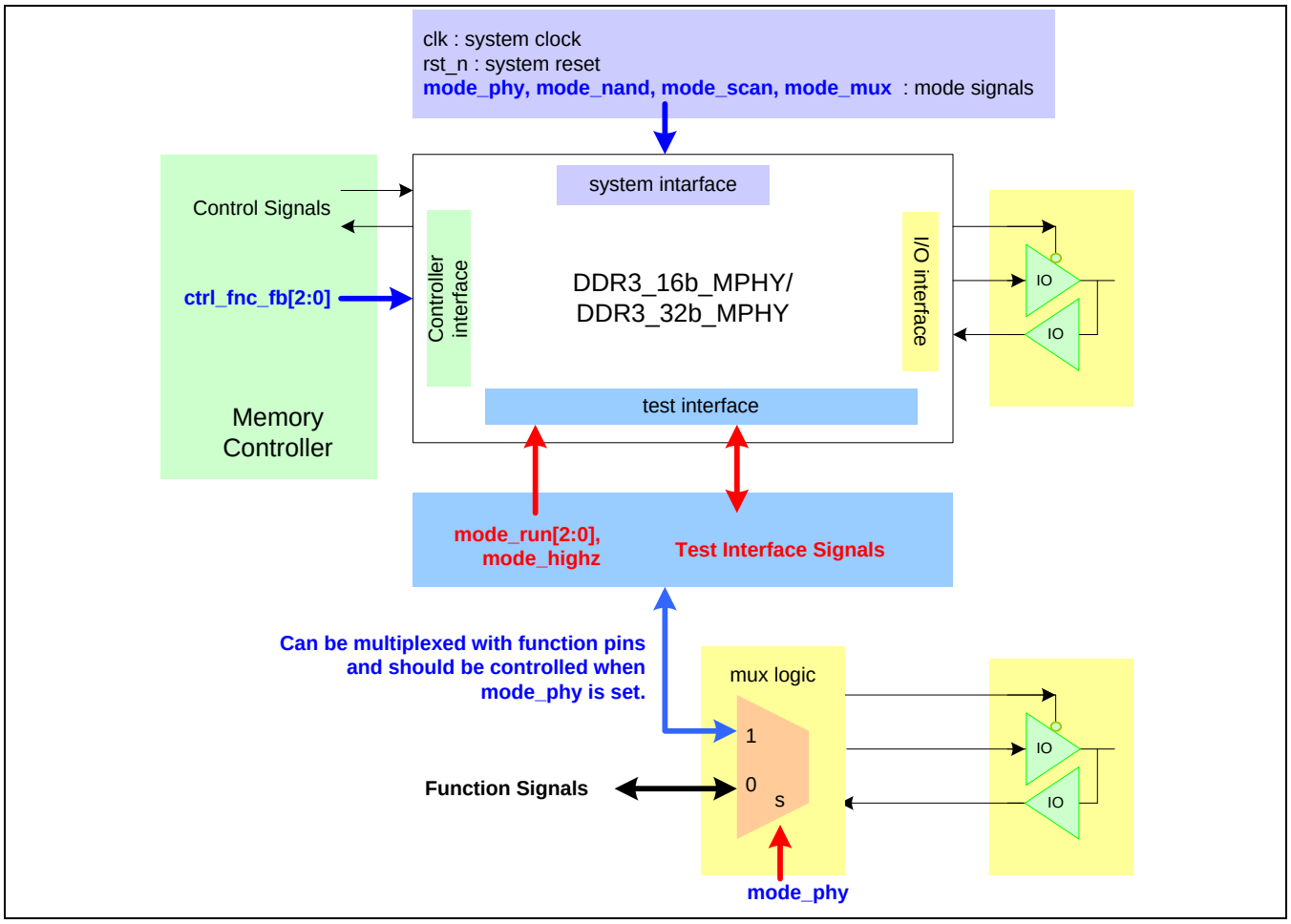

Figure 7-1. Test Pin Allocation

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[2:0]	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:0)	R
	test_ext_offsetr[7:0]	I	Needed for margin test. (Default:8)	R
	test_ext_offsetw[7:0]	I	Needed for margin test. (Default:0)	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
	text_ext_rdlvl_vwmc[31:0]	O	Center value for Valid Window Margin	M
	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. The DLL lock procedure can be removed to decrease a test time in the dll-off mode.

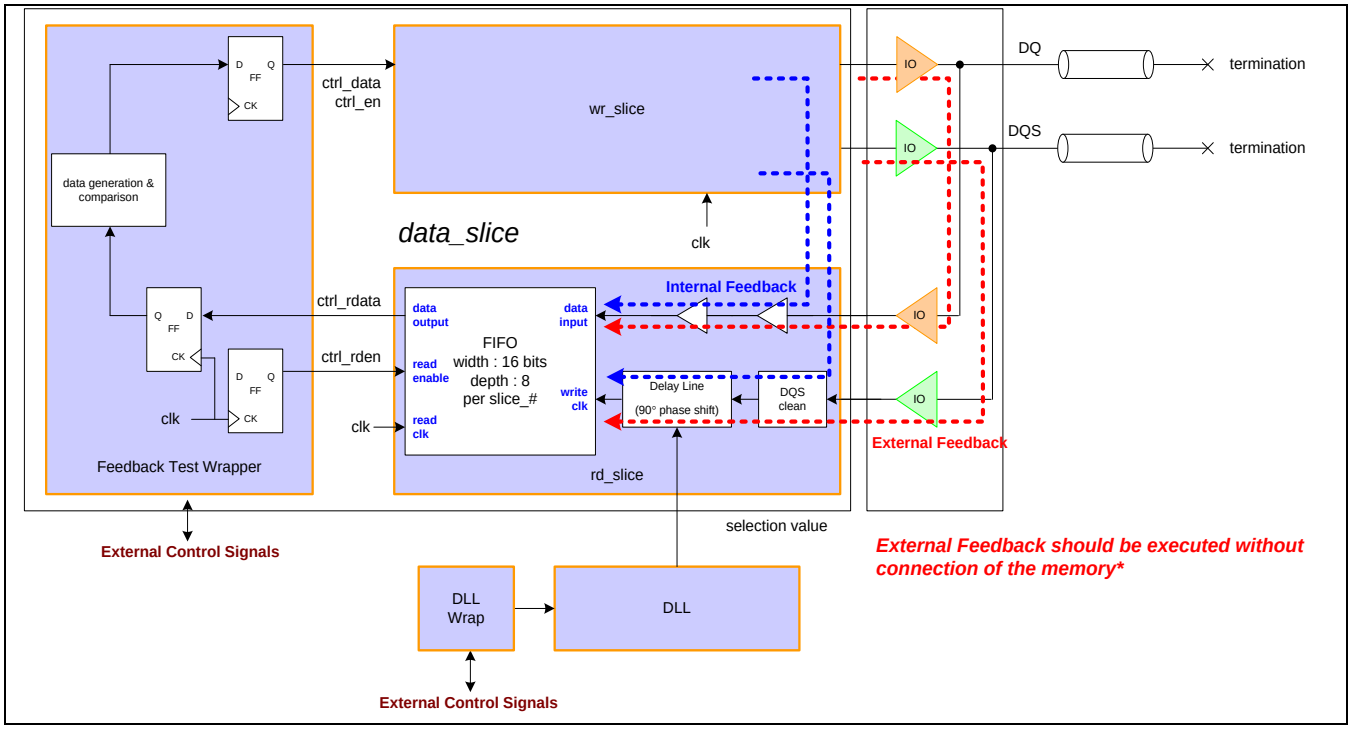

Figure 7-2. PHY Feedback Test Scheme

- 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 for more details)
  - System 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).

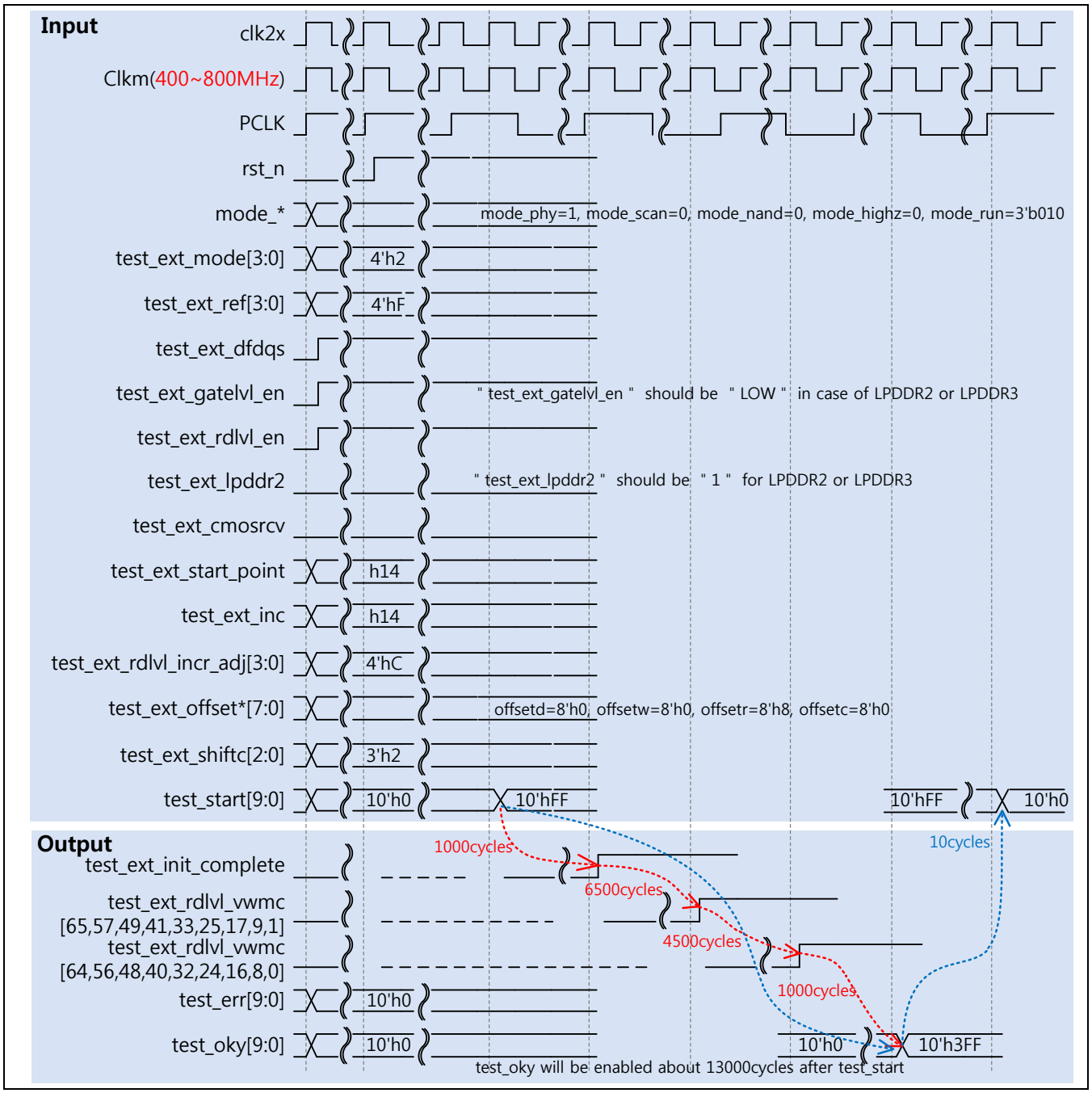

Figure 7-3. PHY Read Feedback

Caution: "dfi_freq_ratio" should be tied as "2'b01" to do feedback test.

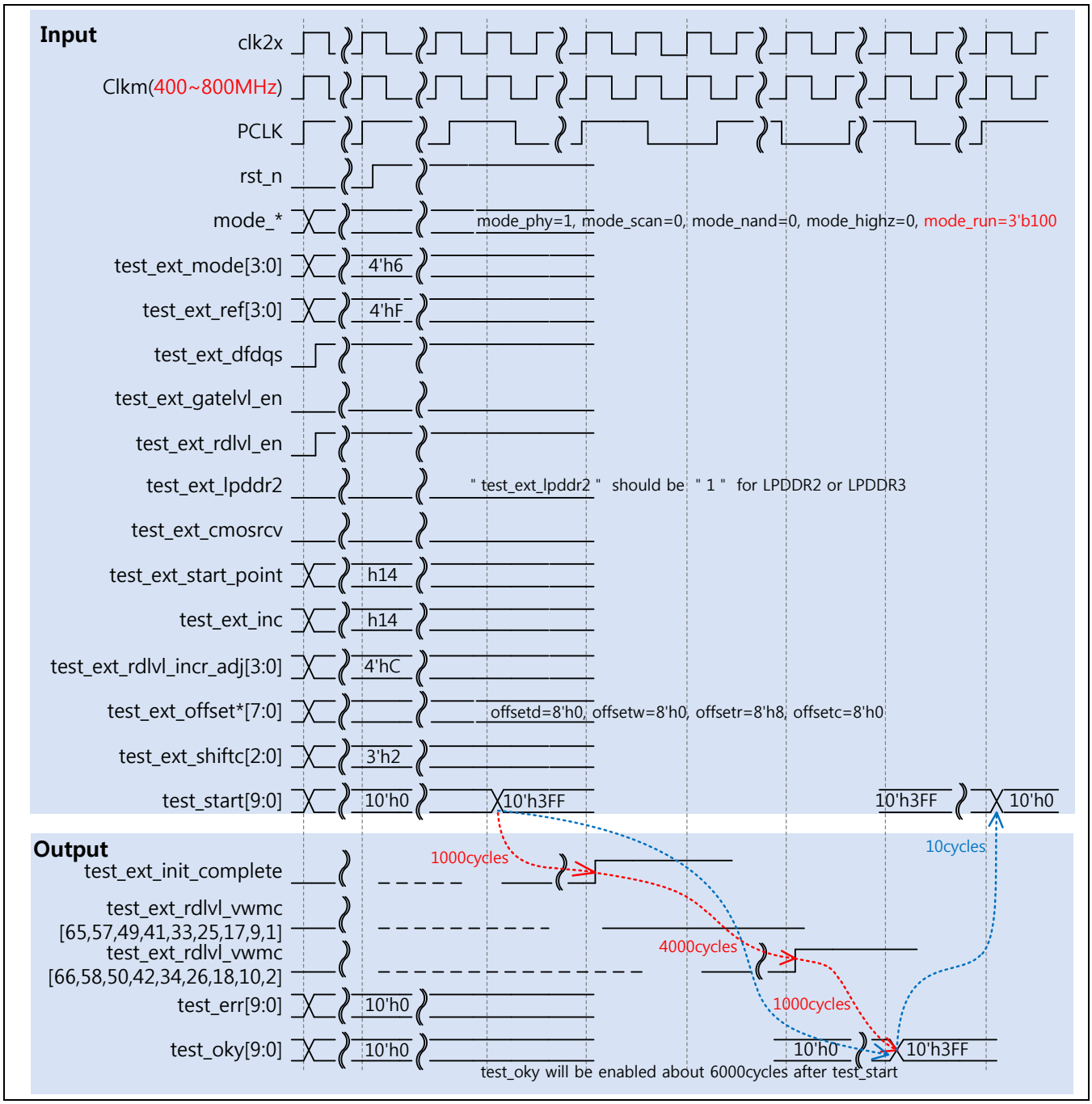

Figure 7-4. PHY Write Feedback

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 are not supported.

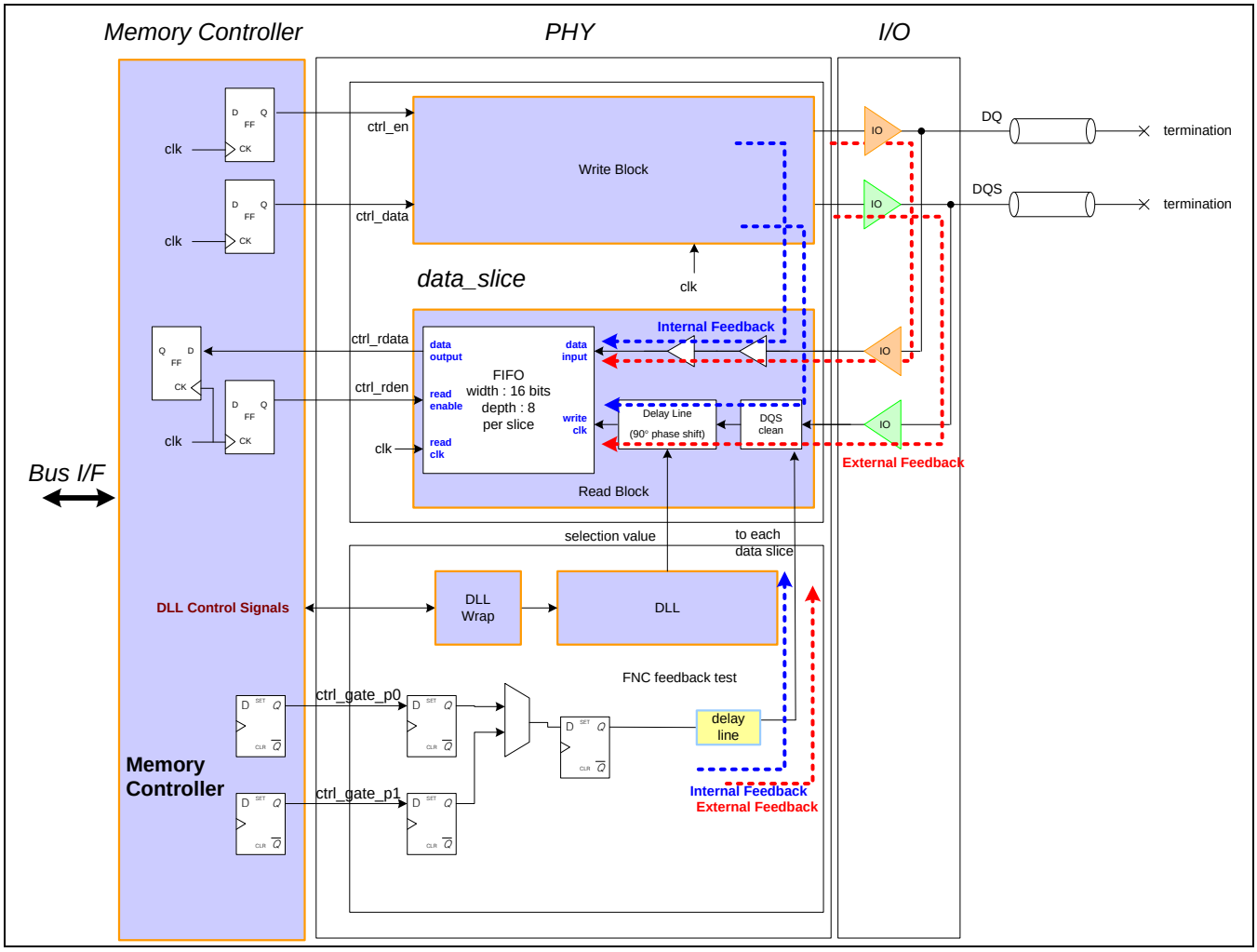

Figure 7-5. FNC Feedback Test Scheme

7.3 ZQ I/O TEST

ZQ I/O should be tested under 400MHz.

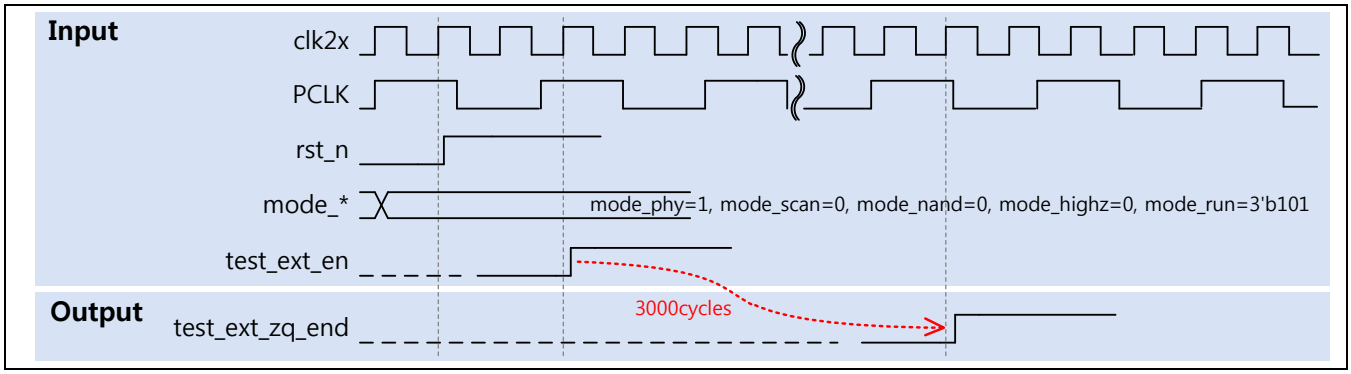

Figure 7-6. The example of ZQ I/O Test

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[24:0], RAS, CAS, WEN, CKE, CK[1:0], GATEO, DM[3:0]

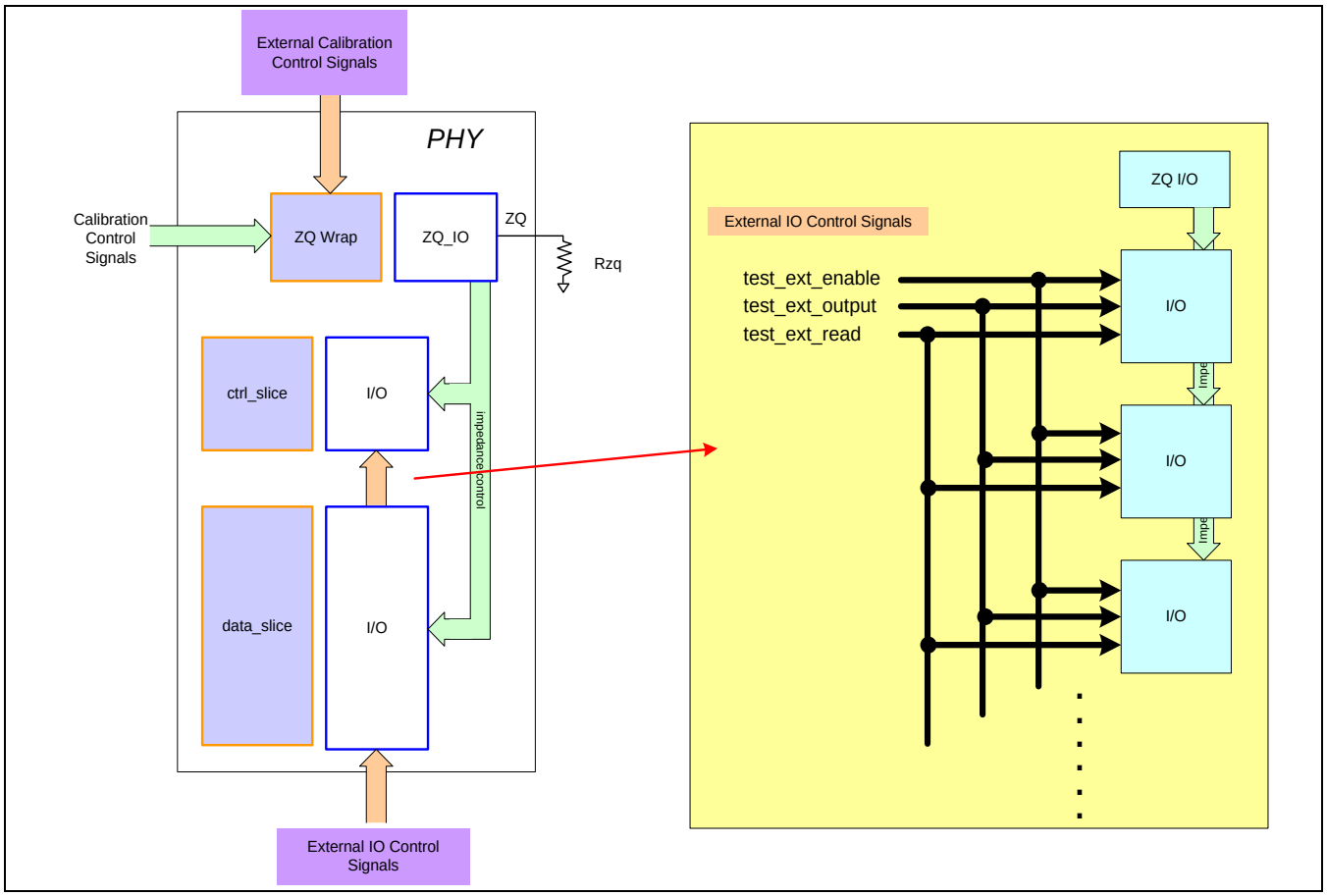

Figure 7-7. I/O Test

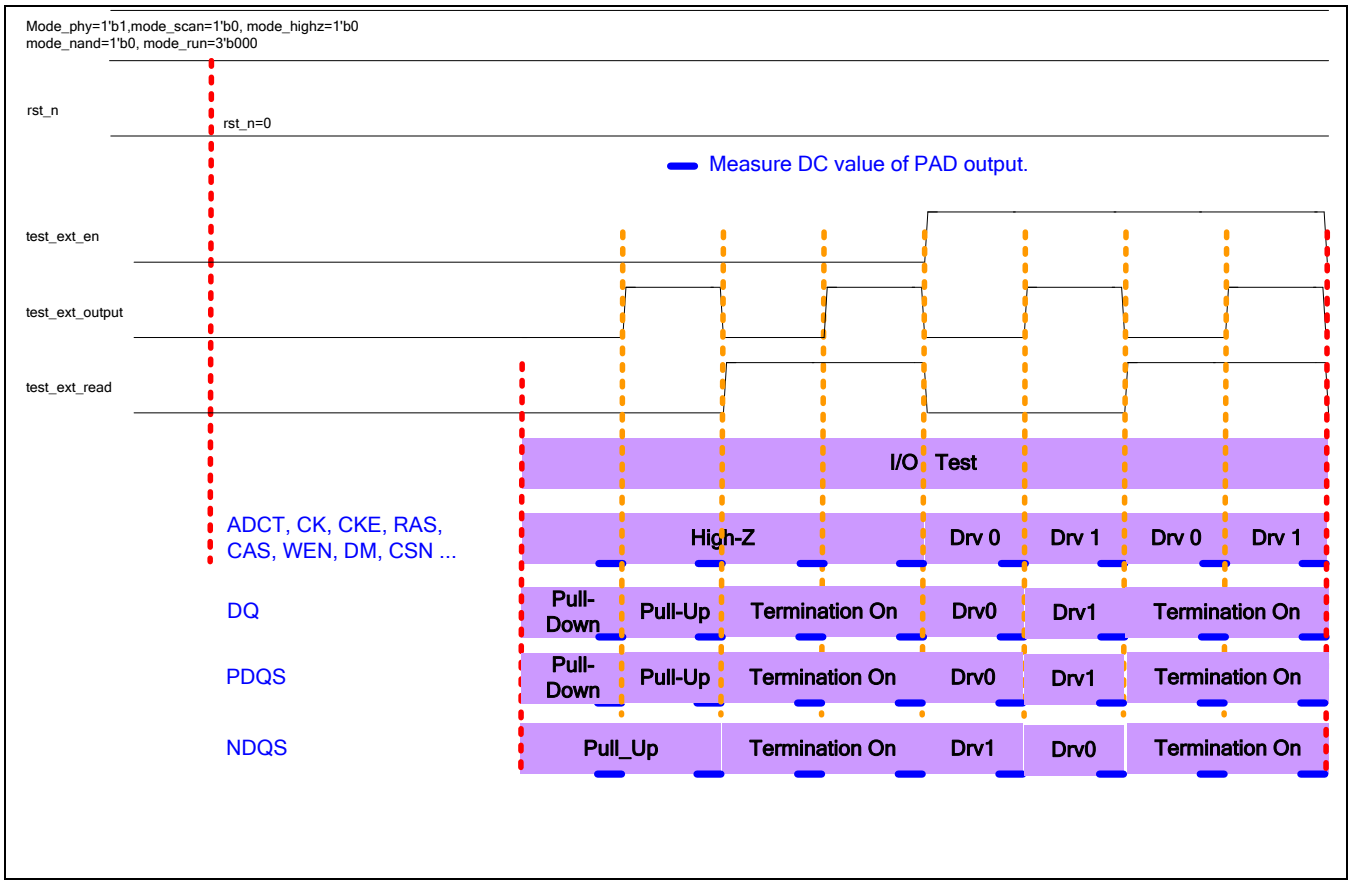

Figure 7-8. The Example of I/O Test

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, DQS except for ADCT, CSN, RAS, CAS, WEN, ODT, CK, CKE, RESET and DM.

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, cmd_active=14'h000E(=LP_DDR_CON3[13:0]),
cmd_default=14'h000F(=LP_DDR_CON4[13:0])
upd_mode=1'b1(=OFFSETD_CON0[28])

- Set Read Latency(RL), Burst Length(BL) and Write Latency(WL)
  - Set RL in PHY_CON4[4:0].
  - Set BL in PHY_CON4[12:8].
  - Set WL in PHY_CON4[20:16].
- ZQ Calibration(Please refer to "8.5 ZQ I/O CONTROL PROCEDURE" for more details)
  - Enable and Disable "zq_clk_div_en" in ZQ_CON0[18]
  - Enable "zq_manual_str" in ZQ_CON0[1]
  - Wait until "zq_cal_done"(ZQ_CON1[0]) is enabled.
  - Disable "zq_manual_str" in ZQ_CON0[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.

Caution: Please don't change the frequency of "clkm" or voltage during operation. Those conditions should be changed without memory access. After changing, "ctrl_start" should be clear and set to lock again.

- Enable DQS pull down mode
  - Set "ctrl_pulld_dqs=9'h1FF" (=LP_CON0[8:0]) in case of using 72bit PHY.
  - Please be careful that DQS pull down can be disabled only after Gate Leveling is done.
- Memory Controller should assert "dfi_ctrlupd_req" after "dfi_init_complete" is set.
  - Please keep "Ctrl-Initiated Update" mode until finishing Leveling and Training.
- Start Memory Initialization by memory controller.
- Skip the following steps if Leveling and Training are not required. (Optional features)
  - 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].
- Recommended that "rdlvl_pass_adj=4" in PHY_CON1[19:16].
- When using DDR3,
  - Set "cmd_active=14'h105E" as default value (=LP_DDR_CON3[13:0])
  - Set "cmd_default=14'h107F" as default value (=LP_DDR_CON4[13:0])
- When using LPDDR2 or LPDDR3,
  - Set "cmd_active=14'h000E" (=LP_DDR_CON3[13:0])
  - Set "cmd_default=14'h000F" (=LP_DDR_CON4[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 MDLL_CON0[5] before Leveling.
  - Read "ctrl_lock_value[8:0]" in MDLL_CON1[16:8].
  - Update "ctrl_force[8:0]" in MDLL_CON0[15:7] by the value of "ctrl_lock_value[8:0]".
- 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).
- Read DQ Calibration(=Read Leveling) (refer to 8.1.4)
- Write Leveling Calibration (refer to 8.1.5)
- 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 T_RDDATA_CON0 after Read DQ Calibration.
- Write DQ Calibration (refer to 8.1.6)
- Set "ctrl_dll_on=1" (=MDLL_CON0[5]).
- Set "DLLDeskewEn=1" (=PHY_CON2[12]) to compensate Voltage, Temperature variation during operation.

Caution: Don't assert "ctrl_ctrlupd_req" during Leveling or Training.

- Set "upd_mode=0" (=OFFSETD_CON0[28]) for PHY-Initiated Update.
  - If Ctrl-Initiated Update is used, set "upd_mode=1" (refer to 8.7)
- Enable and Disable "ctrl_resync"(=OFFSETD_CON0[24])" to make sure All SDLL is updated.

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.

8.1.1.1 H/W Write Leveling

- Memory Controller should configure Memory (DDR3, LPDDR3) in Write Level mode.
  - Set "cmd_default[8:7] =2'b11" (LPDDR_CON4[8:7]) to enable "ODT[1:0]" signals during Write Leveling.
- Configure PHY in Write Level mode.
  - Enable "wrlvl_mode" in PHY_CON0[16].
- Start Write Leveling by setting "wrlvl_start = 1'b1" (=PHY_CON3[16])
- Wait until "wrlvl_resp = 1'b1" (=PHY_CON3[24])
- Finish Write Leveling by setting "wrlvl_start = 1'b0" (=PHY_CON3[16])
- Configure PHY in normal mode.
  - Disable "wrlvl_mode" in PHY_CON0[16].
- Set "cmd_default[8:7]=2'b00" (LPDDR_CON4[8:7]) to disable "ODT[1:0]" signals.
- Set "reg_mode[0]=1'b1"(=PHY_CON3[0]).
- Memory Controller should configure Memory (DDR3, LPDDR3) in normal mode.
- Enable and Disable "ctrl_resync"(=OFFSETD_CON0[24])" to make sure All SDLL is updated.
- Recommend to set "ctrl_readduradj=1"

8.1.1.2 S/W Write Leveling

- Memory Controller should configure Memory (DDR3, LPDDR3) in Write Level mode.
  - Memory Controller should assert "dfi_odt_p0/p1" to enable "ODT[1:0]" 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 p156).
- To find out the optimal Write Level De-skew DLL code for the alignment between CK and DQS
  - Set Write Level code for all data_slice (WR_LVL_CON0, WR_LVL_CON1, WR_LVL_CON2).(1)
    - The start code value should be 0x8.
  - Update SDLL code(WR_LVL_CON0, WR_LVL_CON1, WR_LVL_CON2).(2)
    - Enable "ctrl_wrlvl_resync" (=WR_LVL_CON3[0])
    - Disable " ctrl_wrlvl_resync" (=WR_LVL_CON3[0])
  - Memory Controller should generate 1 cycle pulse of "dfi_wrdata_en_p0".(3)
  - Memory Controller should read the value of "ctrl_io_rdata[8x*]" which is output of PHY.(4)
    - If it is zero, Increment "ctrl_wrlvl_code*" by "1" and then go to "2" to update code for "Data Slice *".
    - If it is one with the start code(=0x8), keep incrementing "ctrl_wrlvl_code*" by "1" until it is zero. Go to "2" to update code for "Data Slice".
    - If it is one with zero at the previous step, "ctrl_wrlvl_code* - 1" will be the optimal code for "Data_Slice *"
    - If the optimal codes for all Data_Slice are searched, go to "5". Otherwise go to "2" to update the incremented codes.

NOTE: "ctrl_io_rdata" can be also read from DQ_IO_RDATA0, DQ_IO_RDATA1 and DQ_IO_RDATA2.(p109)
"*" means 0,1,2,3,4,5,6,7,8 in case of 72bit PHY and 0,1,2,3 in case of 32bit PHY

- Configure PHY in normal mode after 4 are finished.(5)
  - Disable "wrlvl_mode" (=PHY_CON0[16])
- Memory Controller should configure Memory (DDR3, LPDDR3) in normal mode.(6)
- Enable and Disable "ctrl_resync"(=OFFSETD_CON0[24])" to make sure All SDLL is updated.
- Recommend to set "ctrl_readduradj=1"

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.

8.1.1.3 Write Leveling DLL Manual Setting

After knowing the board and package delay exactly, you can use the following manual setting instead of Write Leveling.

- Set WR_LVL_CON0, WR_LVL_CON1 and WR_LVL_CON2 for each data_slice.
  - For example, suppose the following conditions
    - "ctrl_lock_value[8:0] = 0x7F(=127)"
    - The delay of CK is about 118ps, 295ps, 512ns, 704ps, 920ps, 1137ps, 1196ps, 1329ps and 861ps at each DRAM in DIMM.
  - Fine step delay will be about 9.84ps by calculating "1250/127" at 800MHz.
  - The delay of CK can be represented by 0x0C, 0x1E, 0x34, 0x47, 0x5D, 0x73, 0x79, 0x08, and 0x57 with Fine step delay unit.
    - If the delay of CK at DRAM is greater than 1 or 2 cycle, please ignore that cycle delay when setting "WR_LVL_CON*". For example, the delay of CK at 7th DRAM is 1329ns and the fine step delay should be 0x85, but after ignoring 0x7F (=1250ns), the setting value will be 0x08.
  - Please set "WR_LVL_CON0=0x47341E0C", "WR_LVL_CON1=0x0879735D" and "WR_LVL_CON2=0x57"
- Set "wrlvl_mode=1" (=PHY_CON0[16])
- Set "wrlvl_mode=0" (=PHY_CON0[16])
- Enable and Disable "ctrl_resync"(=OFFSETD_CON0[24])" to make sure All SDLL is updated.

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. (=OFFSETD_CON0[7:0])
  - Change CA SDLL code in OFFSETD_CON0[7:0]. (1)
    - The start code value should be 0x8.
  - Update CA SDLL code in OFFSETD_CON0[7:0]. (2)
    - Enable "ctrl_resync" in OFFSETD_CON0[24]

    - Disable " ctrl_resync" in OFFSETD_CON0[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(=OFFSETD_CON0[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 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.

- Enable "gate_lvl_start(=PHY_CON3[18])" to do read leveling.(2)
- Wait until "rd_wr_cal_resp"(=PHY_CON3[26]) 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)".
- Disable "gate_lvl_start(=PHY_CON3[18])" after "rd_wr_cal_resp"(=PHY_CON3[26]) 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. (MR3:A2=1, 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]" or "[0x0000_0000,0xFFFF_FFFF, 0x0000_0000,0xFFFF_FFFF]" in MPR mode.
- 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]" or "[0xFFFF_FFFF,0x0000_0000, 0xFFFF_FFFF, 0x0000_0000]".
  - Set "lpddr2_addr=20'h208"(=LP_DDR_CON0[19:0]) to issue MR32 during DQ Calibration
    - 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]
- Enable "rd_cal_start"(=PHY_CON3[19]) to do read leveling.(2)
- Wait until "rd_wr_cal_resp"(=PHY_CON3[26]) 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)".
- Disable "rd_cal_start"(=PHY_CON3[19]) after "rd_wr_cal_resp"(=PHY_CON3[26]) is enabled.(4)
- In case of using DDR3 Memory,
  - Memory Controller should disable MPR in SDRAM device(MR3:A2=0, Please refer to JEDEC Standard).

8.1.5 WRITE LATENCY CALIBRATION

Write Latency Calibration can adjust Write Latency for each slice after checking if there are some DQS signals delayed than CK. To make sure the read operation, Read DQ Calibration is required before Write Leveling Calibration. When designing Package and Board, the skew between DQS and DQ at each slice should be minimized for Write Leveling Calibration.

- Set Write Latency(=ctrl_wrlat) before Write Latency Calibration.
  - Set "ctrl_wrlat" (=PHY_CON4[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 write and read the pattern in "PHY_CON1[15:0]" to know whether Write Leveling is done normally or not after Read DQ Calibration.(Read DQ Calibration is needed)
  - In case of using LPDDR2 or LPDDR3 memory,
    - The column address should be defined in "lpddr2_addr"(=LP_DDR_CON0[19:0]).
    - "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.
  - In case of using DDR3 memory,
    - The column address should be defined in "ddr3_addr"(=LP_DDR_CON2[15:0])
    - 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" for DQ Calibration in case of BL=4.
- Set "wl_cal_mode=1" (=PHY_CON3[20]).
- Set "wl_cal_start=1" (=PHY_CON3[21]) to do Write Leveling Calibration.
- Wait until "wl_cal_resp" (=PHY_CON3[27]) is set.
- Set "wl_cal_start=0" (=PHY_CON3[21]).

8.1.6 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 "ctrl_wrlat" by "WL" (=PHY_CON4[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 defined in "PHY_CON1[15:0]" according to the following settings.(2)
  - In case of using LPDDR2 or LPDDR3 memory,
    - The column address should be defined in "lpddr2_addr" in LP_DDR_CON0[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 will be 0x5 if the column address is 0x0 and bank address is 0x0. In case of CA swap mode, lpddr2_addr=20'h204 if the column address is 0x0 and bank address is 0x0.
    - 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,
    - The column address should be defined in "ddr3_addr" in LP_DDR_CON2[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" for DQ Calibration in case of BL=4.
    - 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.
- Set Write Training Mode(3)
  - Set "wr_cal_mode=1"(=PHY_CON2[26]).
- Set "wr_cal_start=1" in PHY_CON2[27] to do Write DQ Calibration.(4)
- Wait until "rd_wr_cal_resp(=PHY_CON3[26])" 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)".
- Set "wr_cal_start=0"(=PHY_CON2[27]) after "rd_wr_cal_resp"(=PHY_CON3[26]) 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[8:0] to ctrl_force[8:0].
  - 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 0x1FF, "ctrl_force" will be "0x1FF".
- the low frequency is under 100MHz,
  - "ctrl_force"=0x1FF, "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 = 0x8.
- 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).

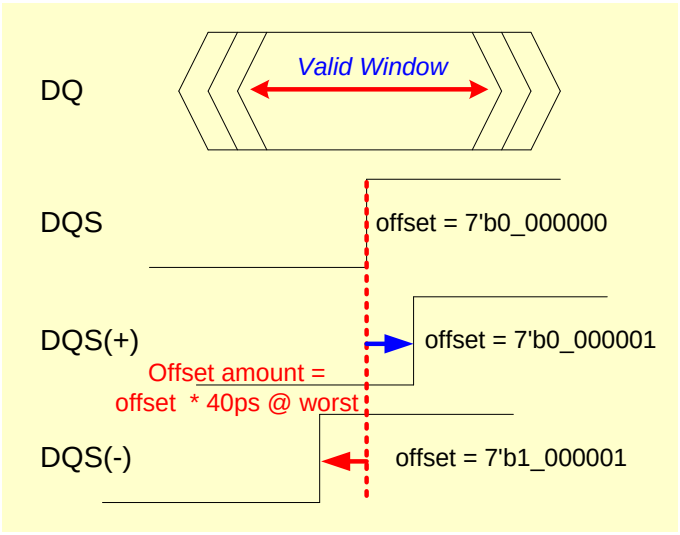

Figure 8-1. Offset Control Example for ctrl_offset0~ctrl_offset3

8.4 DLL LOCK PROCEDURE

- After power-up and system PLL locking time, system reset(rst_n) is released.
- ~~CPU sets ctrl_offsetc*, ctrl_offsetr*, ctrl_offsetw*, ctrl_shiftc*, ctrl_start_point and ctrl_inc value. Default values can be used normally. (If these values or the frequency are changed during the normal operation, the following steps should be applied again after ctrl_dll_on is cleared. "*" means 0, 1, 2, 3.)~~
  - ~~ctrl_offsetc* = 0x0, ctrl_shiftc* = 0x0~~
  - ~~ctrl_offsetr* = 0x0, ctrl_offsetw*=0x0~~
  - ~~ctrl_start_point = 0x10, ctrl_inc = 0x10~~
- 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[8:0] 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

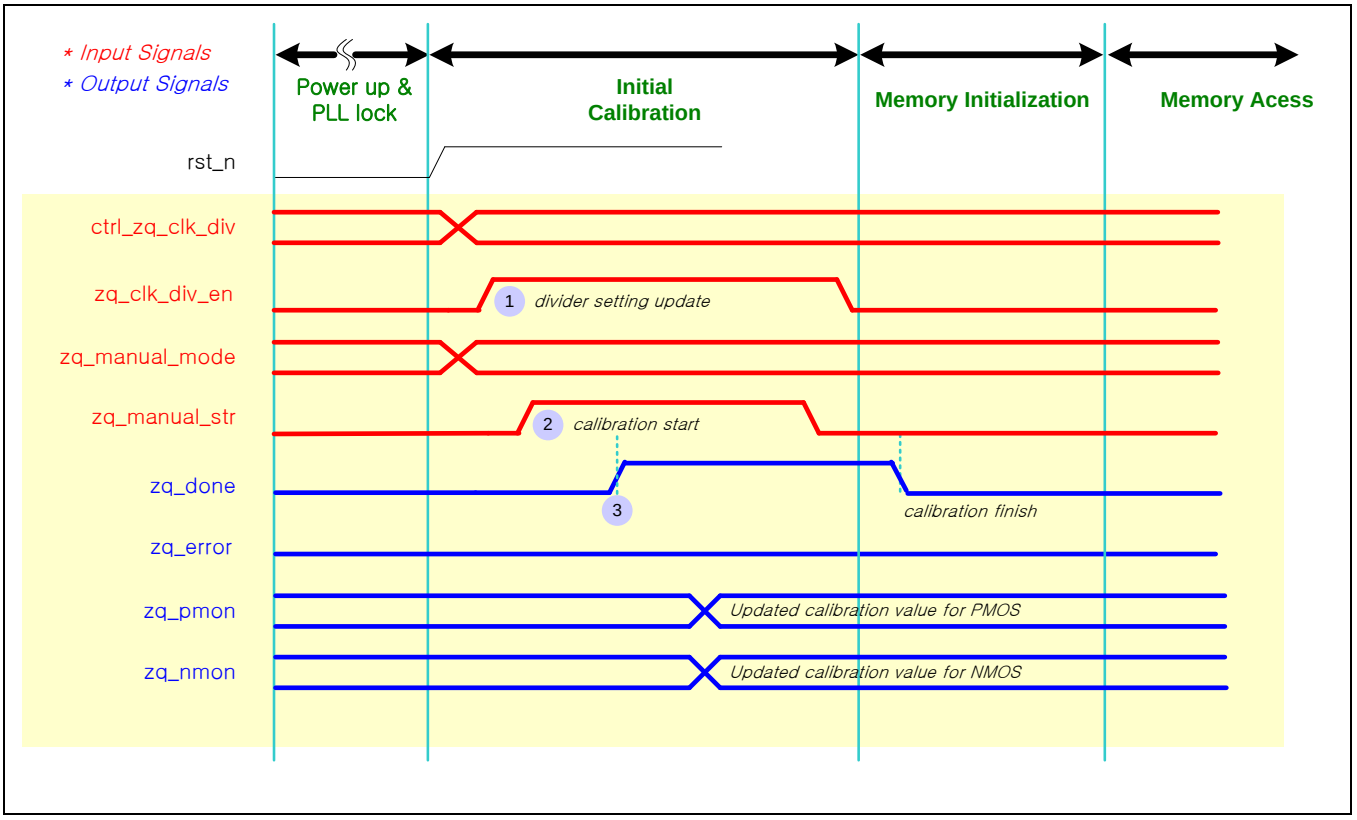

Figure 8-2. One-time calibration procedure

- 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)

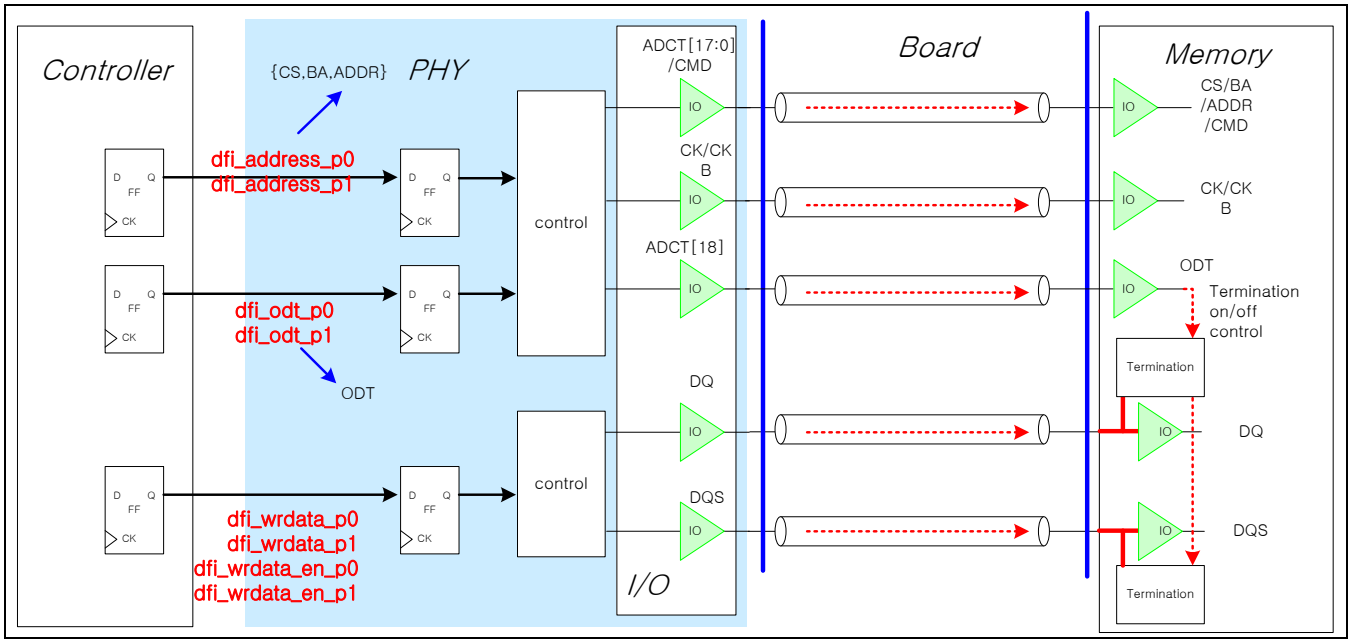

Figure 8-3. Data Write Path and Assignment Example

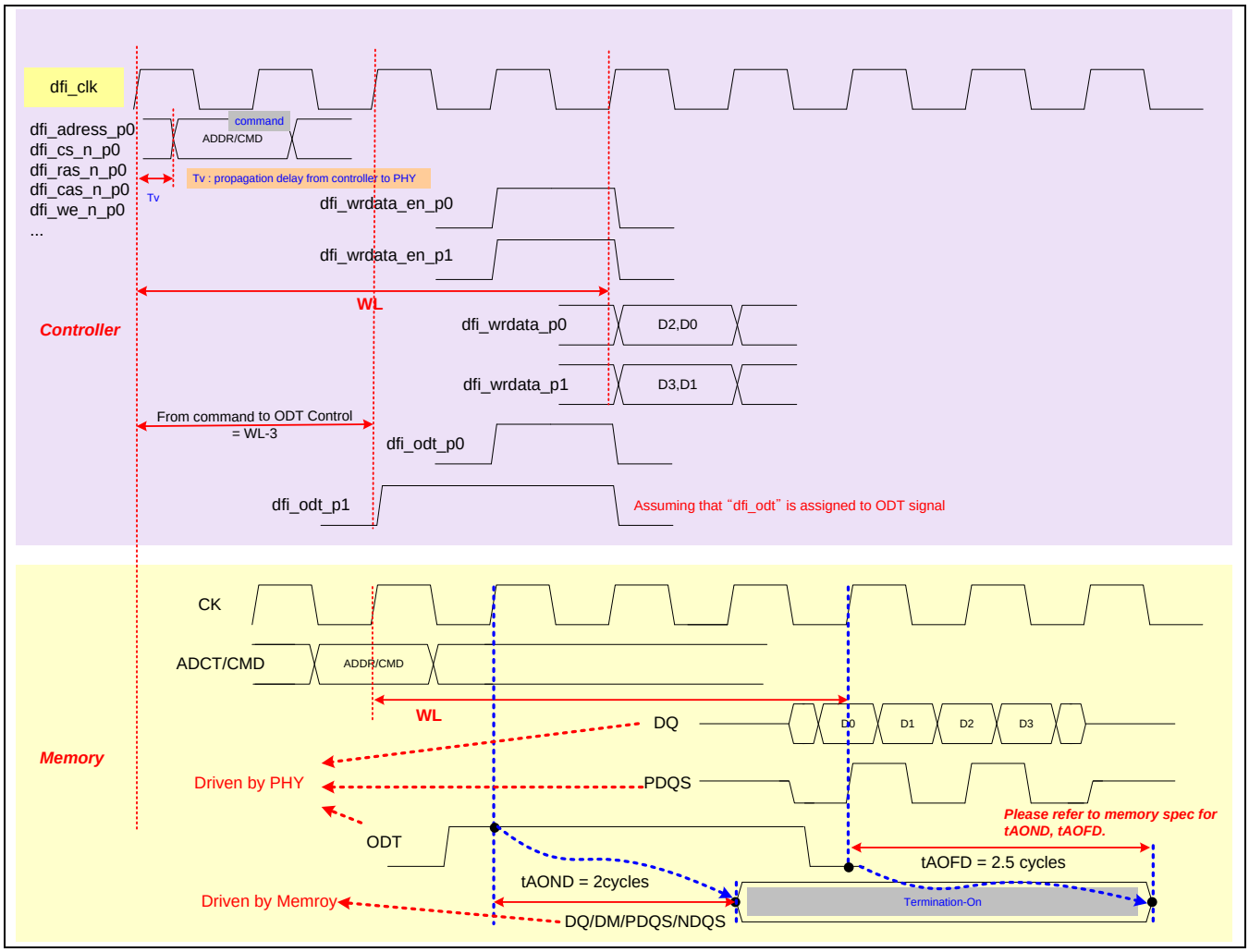

Figure 8-4. Timing Diagram to Control ODT

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 or PHY. 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 p57)

If ctrl_atgate=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 cycles 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.

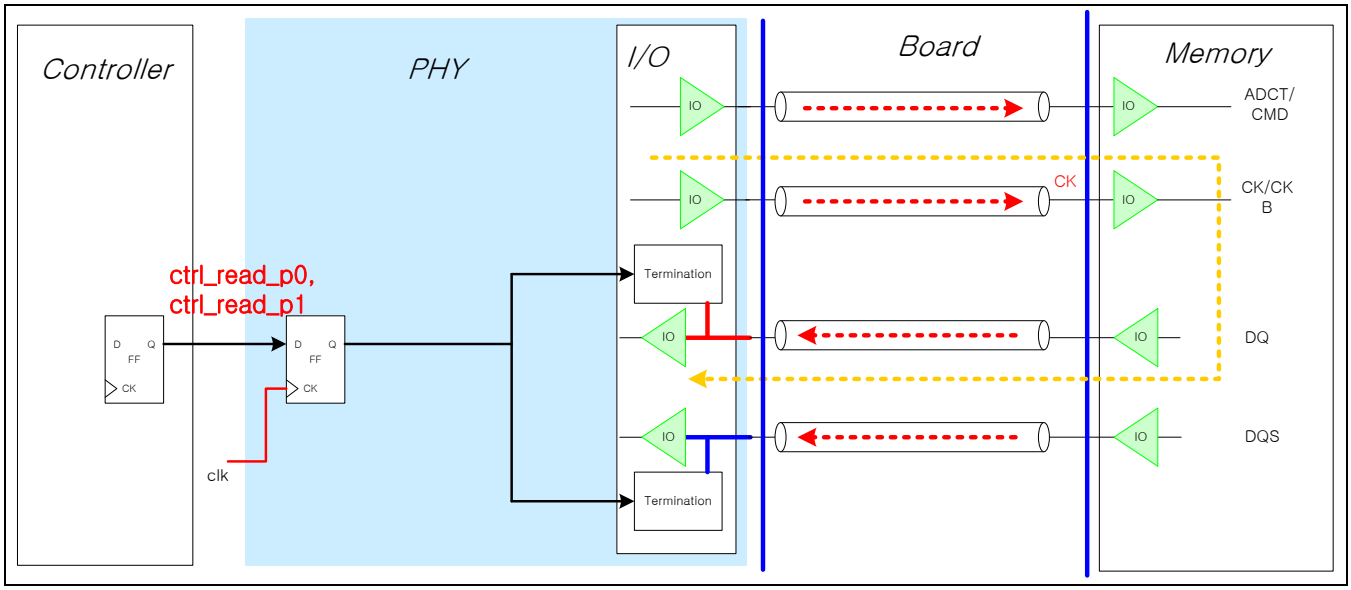

Figure 8-5. Data Read Path

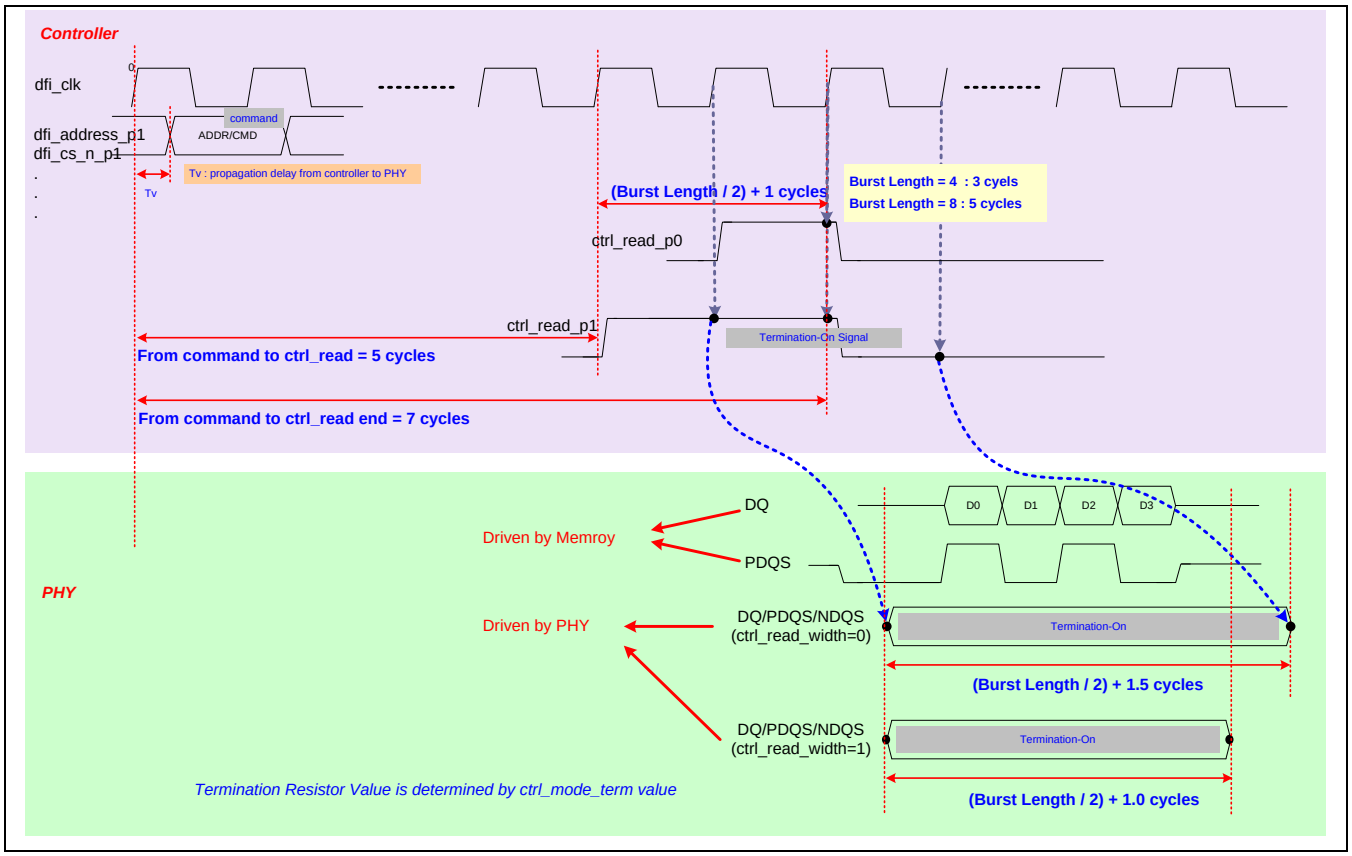

Figure 8-6. Timing Diagram to Control ctrl_read_p0/p1 (Phase1 Read, RL=11)

8.7 DLL CODE UPDATE

Assertion of the "dfi_ctrlupd_ack" or "dfi_phyupd_ack" signal indicates the control, read and write interfaces on the DFI should be idle. While the "dfi_ctrlupd_ack" or "dfi_phyupd_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=21) Please use "MC-Initiated Update" until initialization including all calibrations is finished.

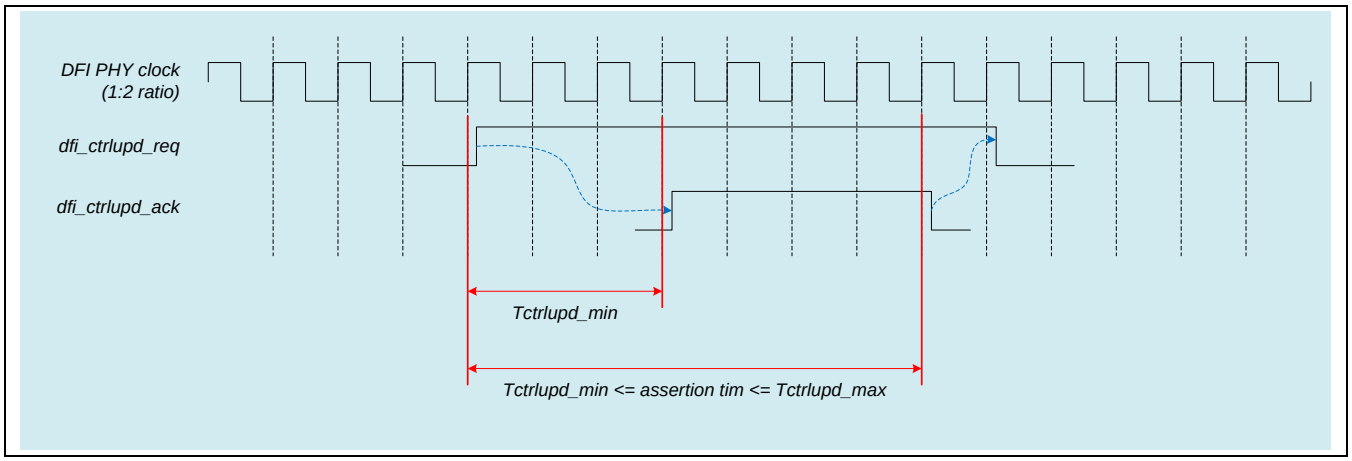

Figure 8-7. MC-Initiated Update Timing Diagram

Assertion of the dfi_phyupd_req signal indicates the control, read and write interfaces on the DFI are required. While the dfi_phyupd_ack signal is asserted, the DFI bus should be idle.

The Tphyupd_typeX parameters indicate the number of cycles of idle time on the DFI control, read and write data interfaces being requested. The dfi_phyupd_ack signal must assert within Tphyupd_resp cycles after the assertion of the dfi_phyupd_req signal. (Tphyupd_resp=64, Tphyupd_typeX=4)

If MC(Memory Controller) doesn't assert dfi_phyupd_ack within Tphyupd_resp, PHY will keep dfi_phyupd_req asserting until dfi_phyupd_ack is asserted and Tphyupd_typeX will be changed up to "21".

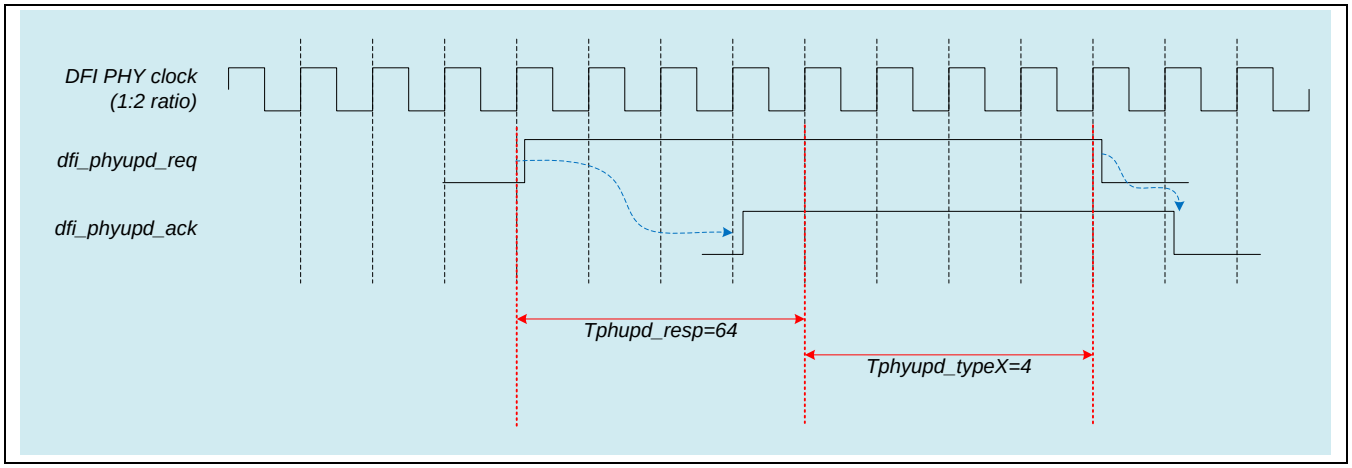

Figure 8-8. PHY-Initiated Update Timing Diagram

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 4 clock cycles after "dfi_phyupd_ack" is issued. So controller should use NOP command with CSN=1 during the update period.

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"(=WR_LVL_CON3[0]) 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 p139)

- CS=1 or CKE=0 if any CA SDLL(LP_DDR_CON2) or DeSkew(WR_LVL_CON*) Code is changed.
- Any Write operation shouldn't be permitted if Write SDLL(OFFSETW_CON*) 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.

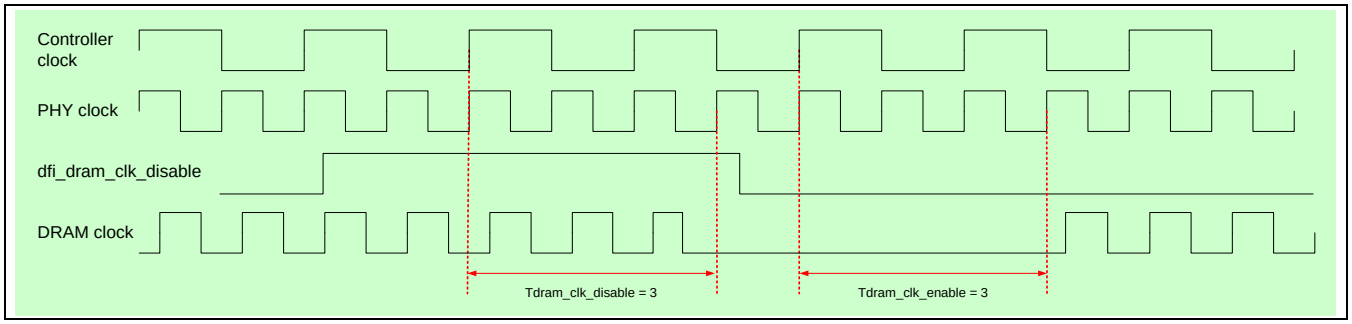

Figure 8-9. Clock Control Timing