



## Synopsys PHY IP

### LPDDR5X/5/4X PHY Utility Block (PUB)

#### Databook

---

*DWC LPDDR5X/5/4X PHY*

## Copyright Notice and Proprietary Information

© 2024 Synopsys, Inc. All rights reserved. This Synopsys software and all associated documentation are proprietary to Synopsys, Inc. and may only be used pursuant to the terms and conditions of a written license agreement with Synopsys, Inc. All other use, reproduction, modification, or distribution of the Synopsys software or the associated documentation is strictly prohibited.

### Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

### Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

### Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at  
<https://www.synopsys.com/company/legal/trademarks-brands.html>

All other product or company names may be trademarks of their respective owners.

### Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

### Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.

[www.synopsys.com](http://www.synopsys.com)

# Contents

Revision History .....	23
<b>Preface .....</b>	<b>37</b>
Databook Organization .....	37
Reference Documentation .....	37
STAR on the Web (SotW) .....	38
Synopsys Statement on Inclusivity and Diversity .....	38
Web Resources .....	38
Customer Support .....	38
<b>Chapter 1</b>	
<b>Product Overview .....</b>	<b>41</b>
1.1 System Overview .....	42
1.2 System Components .....	43
1.3 Firmware/PHYINIT Usage Model .....	45
1.3.1 ATE Firmware Usage Model .....	45
1.3.2 Training Firmware Usage Model .....	45
1.3.3 PHYINIT Usage Model .....	45
1.4 Memory Controller Solution .....	46
1.5 LPDDR5X/5/4X PHY Features .....	47
1.5.1 Highlights .....	47
1.5.2 LPDDR5/5x-Specific Features .....	50
1.5.3 LPDDR4X-Specific Features .....	51
1.5.4 I/O Features .....	52
1.6 Compatible Standards .....	53
1.7 Supported System Configurations .....	54
1.7.1 PHY AC Configuration .....	54
1.7.2 PHY DATA Configuration .....	56
1.8 PUB Area .....	57
1.9 PUB Utilization .....	58
1.10 Timing Closure .....	59
1.11 Limitation and Restrictions .....	60
1.12 Definition of Terms and Acronyms .....	61
<b>Chapter 2</b>	
<b>Functional Overview .....</b>	<b>65</b>
2.1 PUB Functional Overview .....	66
2.2 DFI Connections and System Memory Configurations .....	69

**Chapter 3**

<b>Signal Descriptions</b> .....	73
3.1 DFI Signals .....	74
3.2 Misc Signals .....	83
3.3 Reset Signals .....	84
3.4 PLL Clock Signals .....	85
3.5 Test Signals .....	86
3.6 TDR Signals .....	87
3.7 Scan Signals .....	89
3.8 AHB-Lite Signals .....	91
3.9 APB Signals .....	93
3.10 SRAM Signals .....	95
3.11 Bypass mode Signals .....	103
3.12 Bump Signals .....	109
3.13 Power and Ground Signals .....	111

**Chapter 4**

<b>Signal Mapping</b> .....	113
4.1 DFI Pin List and Pin Mapping .....	114
4.1.1 DBYTE Assignment .....	114
4.1.2 DFI Pin List .....	114
4.1.3 LPDDR5/5X DFI-to-SDRAM Pin Map DWC_LPDDR5XPHY_NUM_CHANNELS_1 .....	114
4.1.4 LPDDR5/5X DFI-to-SDRAM Pin Map DWC_LPDDR5XPHY_NUM_CHANNEL_2 .....	115
4.1.5 LPDDR4X DFI-to-SDRAM Pin Map DWC_LPDDR5XPHY_NUM_CHANNEL_1 .....	118
4.1.6 LPDDR4X DFI-to-SDRAM Pin Map DWC_LPDDR5XPHY_NUM_CHANNEL_2 .....	119
4.2 DFI Pin Swizzling .....	122
4.3 DFI Interface Implementation Details .....	123
4.4 DFI Phase Bits, Chip Selects and Data Bits .....	124
4.5 DFI Command Phase Timing .....	125

**Chapter 5**

<b>Parameter Description</b> .....	127
5.1 PUB Parameters .....	128
5.1.1 DFI Control Timing Parameters .....	128
5.1.2 DFI Write Parameters and Write Timing Parameters .....	129
5.1.3 DFI Write Timing Parameters .....	129
5.1.4 DFI WCK Control Timing Parameters (LPDDR5/5X) .....	131
5.1.5 DFI Read Parameters .....	133
5.1.6 DFI Read Timing Parameters .....	134
5.1.7 DFI Update Timing Parameters .....	135
5.1.8 DFI Low Power Parameters and Low Power Control Timing Parameters .....	136
5.1.9 DFI Error Timing Parameters .....	137
5.1.10 DFI PHY MASTER Timing Parameters .....	138
5.1.11 DFI Frequency Change Parameters .....	138
5.1.12 DFI DRAM Clock Disable Parameters .....	138

**Chapter 6**

<b>Clocks, Resets, Power Domains, Initialization, and Power Management</b> .....	141
6.1 Clocks .....	142
6.1.1 PHY Clocks .....	142
6.1.2 DFx Clocks .....	144
6.2 Resets .....	146

6.2.1 Warm Reset .....	146
6.2.2 Cold Reset .....	147
6.2.3 Reset .....	148
6.2.4 Reset _async .....	148
6.2.5 PRESETn_APB .....	149
6.2.6 WRSTN .....	149
6.3 Power Domains .....	150
6.3.1 BP_PWROK .....	150
6.3.2 Controlled Power off Considerations .....	150
6.4 PHY Initialization .....	151
6.4.1 PHY Initialization Sequence .....	151
6.4.2 Example Timing Diagram for PHY Initialization operation .....	154
6.4.3 Details of Example Timing Diagram of PHY Initialization .....	155
6.5 Power Management .....	157
6.5.1 Active / Mission Mode PHY Power States .....	157
6.5.2 PHY Low Power and Standby States .....	157
6.5.3 DFI Low Power Interface (DFI LP) .....	158
6.5.4 PHY Fast Standby (LP2) .....	159
6.5.5 LP3/IO Retention (LP3) .....	160
6.6 DFI Status Interface .....	169
6.6.1 DFI Initialization .....	169
6.6.2 DFI Frequency Change .....	171
6.6.3 PHY State Changes and dfi_frequency Encoding .....	172
6.6.4 Special Consideration on dfi_freq_fsp (LPDDR5/5x) .....	175
6.6.5 DRAM State During Frequency and Lower Power Changes .....	175
6.6.6 DFI Status Interface Latencies .....	177
<b>Chapter 7</b>	
<b>Architecture .....</b>	<b>185</b>
7.1 Clocking Architecture and Configurations .....	186
7.2 PIPE Architecture Specification .....	189
7.2.1 Compile-time PIPE Overview .....	189
7.2.2 Configurable PIPE Overview .....	191
7.3 Configuration Ports .....	193
7.3.1 APB .....	193
7.3.2 TDR .....	195
7.3.3 Training Sequence .....	197
7.3.4 MailBox .....	197
7.4 Training Support Hardware Overview .....	199
7.4.1 SRAM .....	200
7.4.2 PIE .....	201
7.5 Interrupt Implementation .....	206
7.6 Drift Tracking by MRR Snooping .....	207
7.7 ARC-HS Microcontroller .....	211
7.7.1 ARC-HS Features .....	214
7.7.2 Mapping of ICCM and DCCM in CSR Address Spaces .....	214
7.7.3 ARC SRAM Write Mask Mapping .....	215
7.7.4 ICCM and DCCM Program Loading .....	215
7.7.5 ECC Support .....	216
7.7.6 ARC and SRAM Clocking .....	216

7.7.7 External AHB-Lite Interface . . . . .	217
7.8 PHY PCLK DCA . . . . .	218
7.8.1 Duty Cycle Adjuster in PCLK Receiver . . . . .	218
7.8.2 PCLK Duty Cycle Detection . . . . .	219
7.8.3 PCLK DCA Calibration . . . . .	219
7.8.4 Triggering DCA Calibration for Debug . . . . .	219
7.9 LPDDR5X/5 Specific Features . . . . .	220
7.9.1 Deep Sleep Mode Support . . . . .	220
7.9.2 Write X Function Support . . . . .	220
7.9.3 Link ECC Support . . . . .	220
7.9.4 Strobe-less Read Mode . . . . .	221
7.9.5 Support LPDDR5 Low Frequency . . . . .	222
7.9.6 Single Ended Mode Support . . . . .	222
7.9.7 DRAM ZQ Calibration . . . . .	223
7.9.8 Per-pin DFE Support . . . . .	223
7.10 LPDDR4X Specific Features . . . . .	225
7.10.1 Single Ended Mode Support . . . . .	225
7.10.2 WDQS Extension . . . . .	225
7.11 Impedance Calibration . . . . .	227
7.11.1 External Reference Resistor . . . . .	230
7.11.2 Calibration Engine Setup and CSRs . . . . .	230
7.11.3 Starting Calibration . . . . .	231
7.11.4 Calibration Run Time . . . . .	232
7.11.5 Configuring IO Impedances . . . . .	232
7.12 Low Power Design . . . . .	234
7.13 Support of Dynamic Voltage Scaling . . . . .	235
7.13.1 PHY Core DVFS Support . . . . .	235
7.13.2 DRAM DVFS Support . . . . .	235
7.13.3 DRAM DVFSQ Support . . . . .	235
7.14 Periodic Phase Training 2 (PPT2) . . . . .	237
7.14.1 PPT2 RxEn Retraining . . . . .	237
7.14.2 PPT2 TxDq Wr Retraining . . . . .	237
7.14.3 PPT2 LPDDR5 TxDQ Wr ECC Retraining . . . . .	237
7.14.4 PPT2 RxClk Retraining . . . . .	238
7.14.5 PPT2 Supported Retraining Limits . . . . .	238
7.15 Incremental Retraining Used in PPT2 . . . . .	239
7.15.1 Overview . . . . .	239
7.15.2 Triggering Incremental Retraining . . . . .	239
7.15.3 Determine Retraining Interval . . . . .	240
7.15.4 Determine DFI Control Update Interval . . . . .	241
7.15.5 PHY Retrain Sequencer Options . . . . .	241
7.15.6 Incremental Retraining Limitations . . . . .	241
7.15.7 Incremental Retraining Timing Parameters . . . . .	242
<b>Chapter 8</b>	
<b>System Integration . . . . .</b>	<b>243</b>
8.1 DDR PHY Top Level (DDRPHY_TOP) . . . . .	244
8.1.1 PHY Configuration . . . . .	244
8.2 Additional, Optional Customer Defines . . . . .	246
8.2.1 DWC_LPDDR5XPHY_TOP_PG_PINS Define . . . . .	246

8.2.2 DWC_LPDDR5XPHY_PG_PINS Define .....	246
8.2.3 DWC_LPDDR5XPHY_POP_ENABLED Define .....	246
8.2.4 DWC_LPDDR5XPHY_PCLKCA_RPTR_ENABLED Define .....	246
<b>Chapter 9</b>	
<b>Design for Test .....</b>	<b>249</b>
9.1 Scan Mode .....	250
9.1.1 At Speed Scan Test (ASST) .....	251
9.2 Loopback BIST Mode .....	253
9.2.1 Loopback BIST (ATE) Firmware Setup and Execution .....	253
9.3 Loopback BIST Test Strategy and Hardware .....	254
9.3.1 DQX1 Loopback .....	254
9.3.2 DQSX1 Loopback .....	255
9.3.3 ACX1/CSX1 Loopback .....	255
9.3.4 CKX2 Loopback .....	256
9.3.5 CMOSx1 Loopback .....	256
9.3.6 DTO Loopback .....	256
9.4 Delay Element Testing .....	256
9.4.1 Overview .....	256
9.4.2 Delay Test Hardware .....	257
9.5 Analog/Digital Test Observability .....	258
9.5.1 Digital Observability .....	258
9.5.2 Analog Observability .....	273
9.6 Burn-In (HTOL) Recommendations .....	277
9.7 Bypass Mode .....	278
9.7.1 Bypass Mode Rules .....	280
9.7.2 Bypass Mode Architecture .....	282
9.8 I/O Cell DC Parametric Characterization .....	286
9.8.1 Overview .....	286
9.8.2 VIH/VIL Testing .....	287
9.8.3 VOH/VOL Testing .....	294
9.8.4 ODT Impedance Testing .....	298
9.9 Quiescent Current (IDDQ) Measurement .....	301
9.10 Input Pin Leakage Measurement .....	302
<b>Chapter 10</b>	
<b>Timing .....</b>	<b>303</b>
10.1 PHY Latency .....	304
10.1.1 Max Read Latency (DFIMRL) .....	305
10.1.2 PclkPtrInitVal Impacts From Construction Skews .....	305
10.2 Rank-to-Rank Spacing .....	306
10.3 DQ/DQS Bus Turnarounds .....	307
10.3.1 DRAM Transaction Timing .....	307
10.3.2 PHY Trained State .....	307
10.3.3 LPDDR4X Mode CDD .....	307
10.3.4 LPDDR5 Mode CDD .....	309
10.4 Preamble and Postamble Restrictions .....	311
10.5 PHY Skew Limits .....	312
10.5.1 Skew Parameters .....	314
10.5.2 Write Leveling Delay .....	314
10.5.3 Receive-enable Delay .....	314

10.5.4 DQ-to-DQ Arrival Time Mismatch . . . . .	315
10.5.5 DQ Flight Time to DQS Flight Time Skew . . . . .	315
10.5.6 CS, CKE, Command/Address Flight Time to CK/CK# Flight Time Skew . . . . .	315
10.6 DFI Low Power Interface . . . . .	316
10.6.1 Entering DFI LP State . . . . .	317
10.6.2 Exiting DFI LP State . . . . .	319
10.6.3 DFI LP Data Interface . . . . .	320
10.6.4 DFI LP Control Interface . . . . .	322
10.7 DFI Update Interface . . . . .	323
10.7.1 Controller initiated updates . . . . .	323
10.7.2 PHY Initiated Updates . . . . .	324
10.7.3 PHY Update Interface Timing Parameters . . . . .	325
10.7.4 Collisions Between PHY Update and DFI LP . . . . .	325
10.8 DFI Error Interface . . . . .	326
10.9 DFI Master Interface . . . . .	328
10.9.1 Two-channel Operation . . . . .	328
10.10 PHY DFI Sideband Interactions . . . . .	329
10.10.1 Supported DFI Sideband Interfaces . . . . .	329
10.10.2 Multi-channel Operation . . . . .	330
10.11 DFI Sideband Collisions . . . . .	333
10.11.1 DFI Sideband Simultaneous Collisions . . . . .	333
10.11.2 DFI Sideband Sequential Collisions . . . . .	334
<b>Chapter 11</b>	
<b>Automotive Support . . . . .</b>	<b>337</b>
11.1 LogicBIST (LBIST) . . . . .	338
11.2 Anti-Valent Interrupt . . . . .	339
11.3 Triple Mode Redundant (TMR) Controls . . . . .	340
11.4 Scratch pad register . . . . .	341
11.5 APB Read Parity . . . . .	341
11.6 Static State requirement . . . . .	341
11.7 PLL Lock Indicator . . . . .	342
<b>Chapter 12</b>	
<b>Register Overview . . . . .</b>	<b>343</b>
12.1 Register Map Overview . . . . .	344
12.2 Register Address Overview . . . . .	345
12.3 Specifying Register Addresses . . . . .	346
12.3.1 PState . . . . .	346
12.3.2 Block Type . . . . .	346
<b>Chapter 13</b>	
<b>Register Descriptions . . . . .</b>	<b>349</b>
13.1 DWC_DDRPHYA_DBYTEj_Pk Registers . . . . .	402
13.1.1 DFIMRL_pX (for X = 0; X <= 3) . . . . .	403
13.1.2 EnableWriteLinkEcc_pX (for X = 0; X <= 3) . . . . .	404
13.1.3 DbyteMiscMode . . . . .	405
13.1.4 DxDfiClkDis_pX (for X = 0; X <= 3) . . . . .	406
13.1.5 DxPClkDis_pX (for X = 0; X <= 3) . . . . .	407
13.1.6 LP5DfiDataEnLatency_pX (for X = 0; X <= 3) . . . . .	408
13.1.7 DfiCtrlRxFifoRst . . . . .	409

13.1.8 PptDqsCntInvTrnTg0_pX (for X = 0; X <= 3)	410
13.1.9 PptDqsCntInvTrnTg1_pX (for X = 0; X <= 3)	411
13.1.10 TrackingModeCntrl_pX (for X = 0; X <= 3)	412
13.1.11 RxClkT2UIDlyTg0_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)	415
13.1.12 RxClkT2UIDlyTg1_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)	417
13.1.13 RxClkC2UIDlyTg0_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)	419
13.1.14 RxClkC2UIDlyTg1_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)	420
13.1.15 PptWck2DqoCntInvTrnTg0_pX (for X = 0; X <= 3)	421
13.1.16 PptWck2DqoCntInvTrnTg1_pX (for X = 0; X <= 3)	422
13.1.17 InitSeqControl	423
13.1.18 TxDqsLeftEyeOffsetTg0_pX (for X = 0; X <= 3)	427
13.1.19 MtestMuxSel	428
13.1.20 TxDqsLeftEyeOffsetTg1_pX (for X = 0; X <= 3)	429
13.1.21 RxEnDlyTg0_pX (for X = 0; X <= 3)	430
13.1.22 RxEnDlyTg1_pX (for X = 0; X <= 3)	431
13.1.23 TxDqsRightEyeOffsetTg0_pX (for X = 0; X <= 3)	432
13.1.24 TxDqsRightEyeOffsetTg1_pX (for X = 0; X <= 3)	433
13.1.25 DqsPreambleControl_pX (for X = 0; X <= 3)	434
13.1.26 DbyteRxDqsModeCntrl_pX (for X = 0; X <= 3)	436
13.1.27 RxClkCtl1_pX (for X = 0; X <= 3)	438
13.1.28 TxDqsDlyTg0_pX (for X = 0; X <= 3)	439
13.1.29 TxDqsDlyTg1_pX (for X = 0; X <= 3)	440
13.1.30 TxWckDlyTg0_pX (for X = 0; X <= 3)	441
13.1.31 TxWckDlyTg1_pX (for X = 0; X <= 3)	442
13.1.32 WrLevBits	443
13.1.33 NeverGateDBDlyCalValClk	444
13.1.34 RxClkCtl	445
13.1.35 RxModeCtlRxReplica_pX (for X = 0; X <= 3)	446
13.1.36 RxGainCurrAdjRxReplica_pX (for X = 0; X <= 3)	447
13.1.37 TtcfControl	448
13.1.38 LcdlCalControl	449
13.1.39 NeverGateTrainCntrClk	450
13.1.40 DBYTEParityInvert	451
13.1.41 RdfPtrChkControl	452
13.1.42 DxRxStandbyEn_pX (for X = 0; X <= 3)	454
13.1.43 TxDqLeftEyeOffsetTg0_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)	455
13.1.44 TxDqLeftEyeOffsetTg1_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)	457
13.1.45 TxDqRightEyeOffsetTg0_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)	459
13.1.46 TxDqRightEyeOffsetTg1_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)	461
13.1.47 TrainingCntrSnap_rX (for X = 0; X <= 8)	463
13.1.48 DtsmErrCountSnap_iX (for X = 0; X <= 8)	464
13.1.49 DtsmGoodCountSnap_iX (for X = 0; X <= 8)	465
13.1.50 RxClkTLeftEyeOffsetTg0_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)	466
13.1.51 RxClkTLeftEyeOffsetTg1_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)	468
13.1.52 RxClkTRightEyeOffsetTg0_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)	470
13.1.53 RxClkTRightEyeOffsetTg1_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)	472
13.1.54 RxClkCLeftEyeOffsetTg0_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)	474
13.1.55 RxClkCLeftEyeOffsetTg1_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)	476
13.1.56 RxClkCRightEyeOffsetTg0_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)	478

13.1.57 RxClkCRightEyeOffsetTg1_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8) . . . . .	480
13.1.58 RxFifoVisibility . . . . .	482
13.1.59 RxFifoContents_rX (for X = 0; X <= 8) . . . . .	484
13.1.60 TrainingCntr_rX (for X = 0; X <= 8) . . . . .	485
13.1.61 WckDiffCtl . . . . .	486
13.1.62 PptRxClkInfo_rX (for X = 0; X <= 8) . . . . .	487
13.1.63 RxDigStrbDlyTg0_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8) . . . . .	488
13.1.64 RxDigStrbDlyTg1_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8) . . . . .	490
13.1.65 TxDqDlyTg0_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8) . . . . .	492
13.1.66 TxDqDlyTg1_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8) . . . . .	494
13.1.67 SingleEndedMode_pX (DBYTE) (for X = 0; X <= 3) . . . . .	496
13.1.68 ScratchPadDBYTE . . . . .	497
13.1.69 Dq0LnSel . . . . .	498
13.1.70 Dq1LnSel . . . . .	499
13.1.71 Dq2LnSel . . . . .	500
13.1.72 Dq3LnSel . . . . .	501
13.1.73 Dq4LnSel . . . . .	502
13.1.74 Dq5LnSel . . . . .	503
13.1.75 Dq6LnSel . . . . .	504
13.1.76 Dq7LnSel . . . . .	505
13.1.77 Dq8LnSel . . . . .	506
13.1.78 AsyncDbyteTxMode . . . . .	507
13.1.79 AsyncDbyteRxMode . . . . .	508
13.1.80 AsyncDbyteTxEn . . . . .	509
13.1.81 AsyncDbyteTxData . . . . .	510
13.1.82 AsyncDbyteRxData . . . . .	511
13.1.83 SelDbCurDlyTmngInfo . . . . .	512
13.1.84 DxLoopBackEn . . . . .	513
13.1.85 DxDigStrobeGenSel . . . . .	514
13.1.86 DxDigStrobePat . . . . .	515
13.1.87 DxRxStrobeEnPatWck . . . . .	516
13.1.88 DxOdtEn . . . . .	517
13.1.89 DxPDsampleDIFF . . . . .	518
13.1.90 RxClkTrkErr00 . . . . .	520
13.1.91 PptInfoSel . . . . .	524
13.1.92 Wck2DqoPptInfo . . . . .	525
13.1.93 Dqs2DqPptInfo . . . . .	526
13.1.94 PptRxDqsTrackInfo . . . . .	527
13.1.95 PptRxEnEvnt . . . . .	528
13.1.96 PptCtlStatic . . . . .	529
13.1.97 PptCtlDyn . . . . .	531
13.1.98 RxTrainPattern8BitMode_pX (for X = 0; X <= 3) . . . . .	532
13.1.99 RxTrainPatternEnable . . . . .	533
13.1.100 TrainingParam . . . . .	534
13.1.101 DtsmByteCtrl0 . . . . .	535
13.1.102 DtsmByteCtrl1 . . . . .	536
13.1.103 TxChkDataSelects . . . . .	537
13.1.104 DtsmGateInc . . . . .	538
13.1.105 DtsmGateDec . . . . .	539

13.1.106 DtsmLaneCtrl0_iX (for X = 0; X <= 8)	540
13.1.107 DtsmCmpCount_iX (for X = 0; X <= 8)	541
13.1.108 DtsmErrCount_iX (for X = 0; X <= 8)	542
13.1.109 DtsmGoodCount_iX (for X = 0; X <= 8)	543
13.1.110 DtsmGoodBar	544
13.1.111 DtsmErrBar	545
13.1.112 DtsmCountClears	546
13.1.113 DtsmGoodThldXingInd	547
13.1.114 DtsmErrThldXingInd	548
13.1.115 TrainingIncDecDtsmEn_rX (for X = 0; X <= 8)	549
13.1.116 DqRxVrefDac_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)	550
13.1.117 RxReplicaLcdlPh1UI2UI	551
13.1.118 RxReplicaRatioNow	552
13.1.119 RxReplicaRxClkDlyCorrection	553
13.1.120 RxReplicaStatus00	555
13.1.121 RxReplicaUICalWait	556
13.1.122 RxReplicaInterval	557
13.1.123 RxReplicaDontGateRCTMath	558
13.1.124 DbyteCntrl	559
13.1.125 DbCurrentDlyTimingInfoTgX (for X = 0; X <= 1)	560
13.1.126 DxRxPowerDown	561
13.1.127 RxDigStrbEn_pX (for X = 0; X <= 3)	562
13.1.128 DxPipeEn_pX (for X = 0; X <= 3)	564
13.1.129 LcdlCalCtrl	565
13.1.130 PclkDCDCtrl_pX (DBYTE) (for X = 0; X <= 3)	566
13.1.131 PPTTrainSetup2_pX (for X = 0; X <= 3)	567
13.1.132 ForceInternalUpdate	568
13.1.133 ForceRxDataFifoUpd	569
13.1.134 DMIPinPresent_pX (for X = 0; X <= 3)	570
13.1.135 InhibitTxRdPtrInit_pX (for X = 0; X <= 3)	571
13.1.136 AllowInhibitTxRdPtrInit	572
13.1.137 RdfPtrChkStatusWptrT	573
13.1.138 RDqRDqsCntrl_pX (for X = 0; X <= 3)	574
13.1.139 RxReplicaLcdlCalCtrl	576
13.1.140 RxReplicaRangeVal_pX (for X = 0; X <= 3)	577
13.1.141 RxReplicaCtl04_pX (for X = 0; X <= 3)	578
13.1.142 RdfPtrChkStatusWptrC	580
13.1.143 RxReplicaPathPhase0_pX (for X = 0; X <= 3)	581
13.1.144 RxReplicaPathPhase1_pX (for X = 0; X <= 3)	582
13.1.145 RxReplicaPathPhase2_pX (for X = 0; X <= 3)	583
13.1.146 RxReplicaPathPhase3_pX (for X = 0; X <= 3)	584
13.1.147 RxReplicaPathPhase4_pX (for X = 0; X <= 3)	585
13.1.148 RxReplicaCtl00	586
13.1.149 RxReplicaCtl01_pX (for X = 0; X <= 3)	587
13.1.150 RxReplicaCtl02_pX (for X = 0; X <= 3)	588
13.1.151 RxReplicaCtl03_pX (for X = 0; X <= 3)	589
13.1.152 RdfPtrChkStatusRptr	590
13.1.153 PclkDCACalCtrl0DB	591
13.1.154 PclkDCADynCtrl	592

13.1.155 PclkDCAStaticCtrl0DB_pX (for X = 0; X <= 3)	593
13.1.156 PclkDCASampCntDB	595
13.1.157 PclkDCAHysMaskDB	596
13.1.158 PclkDCACalFineBoundDB	597
13.1.159 PclkDCANextFineOnCoarseDB	598
13.1.160 PclkDCAFullSearchIVACDB	599
13.1.161 PclkDCASampDelayLCDLDB_pX (for X = 0; X <= 3)	600
13.1.162 LcdITstCtrl	601
13.1.163 DtsmErrCountNZ	602
13.1.164 PclkDCAStaticCtrl1DB_pX (for X = 0; X <= 3)	603
13.1.165 PclkDCACoarseBoundDB	604
13.1.166 PclkDCAMiscCtrlIDB	605
13.1.167 TrainingResultsSnap	606
13.1.168 PpgcChkCtrl	607
13.1.169 PpgcChkDbiCtrl	608
13.1.170 PpgcChkDbiConfig	609
13.1.171 PpgcChkLaneMuxSel0	610
13.1.172 PpgcChkLaneMuxSel1	611
13.1.173 PpgcChkTxManipulationCtrl0	612
13.1.174 PpgcChkTxManipulationCtrl1	613
13.1.175 PpgcChkTxManipulationCtrl2	614
13.1.176 PpgcChkTxManipulationPrbs9	615
13.1.177 PpgcChkRxManipulationCtrl0	616
13.1.178 PpgcChkRxManipulationCtrl1	617
13.1.179 PpgcChkRxManipulationCtrl2	618
13.1.180 PpgcChkRxManipulationPrbs9	619
13.1.181 PclkDCAC1kGaterEnDB	620
13.1.182 Prbs0ChkModeSel	621
13.1.183 Prbs0ChkUiMuxSel	622
13.1.184 Prbs0ChkTapDly0	623
13.1.185 Prbs0ChkTapDly1	624
13.1.186 Prbs0ChkTapDly2	625
13.1.187 Prbs0ChkTapDly3	626
13.1.188 Prbs0ChkTapDly4	627
13.1.189 Prbs0ChkTapDly5	628
13.1.190 Prbs0ChkTapDly6	629
13.1.191 Prbs0ChkTapDly7	630
13.1.192 Prbs0ChkStateLo	631
13.1.193 Prbs0ChkStateHi	632
13.1.194 Prbs1ChkModeSel	633
13.1.195 Prbs1ChkUiMuxSel	634
13.1.196 Prbs1ChkTapDly0	635
13.1.197 Prbs1ChkTapDly1	636
13.1.198 Prbs1ChkTapDly2	637
13.1.199 Prbs1ChkTapDly3	638
13.1.200 Prbs1ChkTapDly4	639
13.1.201 Prbs1ChkTapDly5	640
13.1.202 Prbs1ChkTapDly6	641
13.1.203 Prbs1ChkTapDly7	642

13.1.204 Prbs1ChkStateLo .....	643
13.1.205 Prbs1ChkStateHi .....	644
13.1.206 Prbs2ChkModeSel .....	645
13.1.207 Prbs2ChkUiMuxSel .....	646
13.1.208 Prbs2ChkTapDly0 .....	647
13.1.209 Prbs2ChkTapDly1 .....	648
13.1.210 Prbs2ChkTapDly2 .....	649
13.1.211 Prbs2ChkTapDly3 .....	650
13.1.212 Prbs2ChkTapDly4 .....	651
13.1.213 Prbs2ChkTapDly5 .....	652
13.1.214 Prbs2ChkTapDly6 .....	653
13.1.215 Prbs2ChkTapDly7 .....	654
13.1.216 Prbs2ChkStateLo .....	655
13.1.217 Prbs2ChkStateHi .....	656
13.1.218 PpgcChkFltCfg0 .....	657
13.1.219 PpgcChkFltCfg1 .....	658
13.1.220 PpgcChkFltCfg2 .....	659
13.1.221 PpgcChkMskPat0 .....	660
13.1.222 PpgcChkMskPat1 .....	661
13.1.223 PpgcChkMskPat2 .....	662
13.1.224 PpgcChkMskPat3 .....	663
13.2 DWC_DDRPHYA_MASTERj_Pk Registers .....	664
13.2.1 DfiFreqRatio_pX (for X = 0; X <= 3) .....	665
13.2.2 PubDxPowerDownControl .....	666
13.2.3 PclkPtrInitVal_pX (for X = 0; X <= 3) .....	667
13.2.4 CmdFifoWrModeMaster_pX (for X = 0; X <= 3) .....	668
13.2.5 MTestDtoCtrl .....	669
13.2.6 PipeCtl_pX (for X = 0; X <= 3) .....	670
13.2.7 LpDqPhaseDisable .....	671
13.2.8 PubDbyteDisable .....	672
13.2.9 CPclkDivRatio_pX (for X = 0; X <= 3) .....	673
13.2.10 PipeNetDis .....	675
13.2.11 MiscPipeEn .....	676
13.2.12 MtestMuxSelCMOS .....	677
13.2.13 EnRxDqsTracking_pX (for X = 0; X <= 3) .....	678
13.2.14 MtestMuxSel .....	680
13.2.15 MASTERParityInvert .....	681
13.2.16 DfiMode .....	682
13.2.17 MtestPgmInfo .....	683
13.2.18 PhyTID .....	684
13.2.19 DbyteRxEnTrain .....	685
13.2.20 PUBMODE .....	686
13.2.21 DllTrainParam_pX (for X = 0; X <= 3) .....	687
13.2.22 DllControl .....	688
13.2.23 PulseDllUpdatePhase .....	689
13.2.24 ScratchPadMASTER .....	690
13.2.25 PState .....	691
13.2.26 RxFifoInit .....	692
13.2.27 ClockingCtrl .....	693

13.2.28 PhyPipeConfig .....	694
13.2.29 PhyConfig .....	696
13.2.30 LP5Mode .....	699
13.2.31 ForceClkGaterEnables .....	700
13.2.32 D5Mode .....	702
13.2.33 PHYREV .....	703
13.2.34 TxRdPtrInit .....	704
13.2.35 MASTERReservedX (for X == 0) .....	705
13.2.36 PUBReservedP1_pX (MASTER) (for X = 0; X <= 3) .....	706
13.2.37 PclkGateControl .....	707
13.3 DWC_DDRPHYA_ACj_Pk Registers .....	709
13.3.1 AcPipeEn_pX (for X = 0; X <= 3) .....	710
13.3.2 CKDlLStopCal .....	711
13.3.3 CkVal_pX (for X = 0; X <= 3) .....	712
13.3.4 CkDisVal_pX (for X = 0; X <= 3) .....	713
13.3.5 DTOBypassEn .....	714
13.3.6 ACSingleEndedMode_pX (for X = 0; X <= 3) .....	715
13.3.7 InitSeqControl .....	716
13.3.8 MtestMuxSel .....	720
13.3.9 PorControl .....	721
13.3.10 ClrPORMemReset .....	722
13.3.11 MemResetL .....	723
13.3.12 CMOSxHardMacroModeSel .....	724
13.3.13 RxAcVrefControl_pX (for X = 0; X <= 3) .....	725
13.3.14 MemResetLStatus .....	726
13.3.15 ACDlyScaleGatingDisable .....	727
13.3.16 LcdlCalControl .....	728
13.3.17 ACParityInvert .....	729
13.3.18 ACpulseDllUpdatePhase .....	730
13.3.19 ScratchPadAC .....	731
13.3.20 ATestMode .....	732
13.3.21 AcDigStrobeGenSel .....	734
13.3.22 AcDigStrobePat .....	735
13.3.23 AcOdtEn .....	736
13.3.24 AcRxStrobeEnPat .....	737
13.3.25 ATxDlySelect_pX (for X = 0; X <= 3) .....	738
13.3.26 MapCA0toDfi .....	739
13.3.27 MapCA1toDfi .....	740
13.3.28 MapCA2toDfi .....	741
13.3.29 MapCA3toDfi .....	742
13.3.30 MapCA4toDfi .....	743
13.3.31 MapCA5toDfi .....	744
13.3.32 MapCA6toDfi .....	745
13.3.33 AsyncAcTxMode .....	746
13.3.34 AsyncAcRxMode .....	747
13.3.35 AsyncAcTxEn .....	748
13.3.36 AsyncAcTxData .....	749
13.3.37 AsyncAcRxData .....	750
13.3.38 ForceClkDisable .....	751

13.3.39 CaBusTriEn .....	752
13.3.40 AcLnDisable .....	753
13.3.41 DfiClkAcLnDis .....	754
13.3.42 PClkAcLnDis .....	755
13.3.43 AcLcdlCalPhDetOut .....	756
13.3.44 ACXTxDly_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 9) .....	757
13.3.45 CKXTxDly_pX (for X = 0; X <= 3) .....	758
13.3.46 ACXTxDlyDTO_pX (for X = 0; X <= 3) .....	759
13.3.47 ACXTxDly2nMode_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 9) .....	760
13.3.48 AcLcdlUpdInterval_pX (for X = 0; X <= 3) .....	761
13.3.49 LcdlCalSeqUpdCK_pX (for X = 0; X <= 3) .....	762
13.3.50 CkTriEn .....	763
13.3.51 ACReservedX (for X == 0) .....	764
13.3.52 LcdlCalCtrl .....	765
13.3.53 PUBReservedP1_pX (AC) (for X = 0; X <= 3) .....	766
13.3.54 PclkDCDCtrl_pX (AC) (for X = 0; X <= 3) .....	767
13.3.55 ForceInternalUpdate .....	768
13.3.56 ATestPrbsErrCntSECX (for X = 0; X <= 1) .....	769
13.3.57 AcPDsampleSECX (for X = 0; X <= 1) .....	770
13.3.58 AcPDsampleDIFF .....	771
13.3.59 ATestPrbsErrCntDIFF0T .....	772
13.3.60 ATestPrbsErrCntDIFF0C .....	773
13.3.61 PclkDCACalCtrl0AC .....	774
13.3.62 PclkDCADynCtrl .....	775
13.3.63 PclkDCAStaticCtrl0AC_pX (for X = 0; X <= 3) .....	776
13.3.64 PclkDCASampCntAC .....	778
13.3.65 PclkDCAHysMaskAC .....	779
13.3.66 PclkDCACalFineBoundAC .....	780
13.3.67 PclkDCANextFineOnCoarseAC .....	781
13.3.68 PclkDCAFulSearchIVACAC .....	782
13.3.69 LcdlTstCtrl .....	783
13.3.70 AcLoopBackEnLnX (for X = 0; X <= 10) .....	784
13.3.71 ATestPrbsErrCntSEX (for X = 0; X <= 7) .....	785
13.4 DWC_DDRPHYA_PPGCj_Pk Registers .....	786
13.4.1 PpgcGenCtrl .....	787
13.4.2 PpgcGenDbiCtrl .....	788
13.4.3 PpgcGenDbiConfig .....	789
13.4.4 PpgcGenLaneMuxSelX (for X = 0; X <= 1) .....	790
13.4.5 EnPhyUpdZQCalUpdate .....	791
13.4.6 BlockDfiInterface .....	792
13.4.7 BlockDfiInterfaceStatus .....	793
13.4.8 DfiCustMode_pX (for X = 0; X <= 3) .....	794
13.4.9 HwtMRL_pX (for X = 0; X <= 3) .....	795
13.4.10 RegRet .....	796
13.4.11 DisableZQupdateOnSnoop .....	797
13.4.12 Prbs0GenModeSel .....	798
13.4.13 Prbs0GenUiMuxSel .....	799
13.4.14 Prbs0GenTapDlyX (for X = 0; X <= 7) .....	800
13.4.15 MtestMuxSel .....	801

13.4.16 Prbs0GenStateLo .....	802
13.4.17 Prbs0GenStateHi .....	803
13.4.18 Prbs1GenModeSel .....	804
13.4.19 Prbs1GenUiMuxSel .....	805
13.4.20 Prbs1GenTapDlyX (for X = 0; X <= 7) .....	806
13.4.21 Prbs1GenStateLo .....	807
13.4.22 Prbs1GenStateHi .....	808
13.4.23 Prbs2GenModeSel .....	809
13.4.24 Prbs2GenUiMuxSel .....	810
13.4.25 Prbs2GenTapDlyX (for X = 0; X <= 7) .....	811
13.4.26 Prbs2GenStateLo .....	812
13.4.27 Prbs2GenStateHi .....	813
13.4.28 PPTTrainSetup_pX (for X = 0; X <= 3) .....	814
13.4.29 PhyMstrFreqOverride_pX (for X = 0; X <= 3) .....	816
13.4.30 DfiInitComplete .....	817
13.4.31 PPGCParityInvert .....	818
13.4.32 PMIEnable .....	819
13.4.33 Dfi0Status .....	820
13.4.34 Dfi1Status .....	821
13.4.35 DfiHandshakeDelays0_pX (for X = 0; X <= 3) .....	822
13.4.36 DFIPHYUPDX (for X = 0; X <= 1) .....	823
13.4.37 DfiLpCtrlEnX (for X = 0; X <= 1) .....	825
13.4.38 DfiLpDataEnX (for X = 0; X <= 1) .....	826
13.4.39 DynOdtEnCntrlX (for X = 0; X <= 1) .....	827
13.4.40 DfiRespHandshakeDelays0_pX (for X = 0; X <= 3) .....	828
13.4.41 HwtLpCsEnA .....	829
13.4.42 HwtLpCsEnB .....	830
13.4.43 HwtCtrl .....	831
13.4.44 HwtControlOvr .....	832
13.4.45 ScratchPadPPGC .....	833
13.4.46 HwtControlVal .....	834
13.4.47 ForceHWTClkGaterEnables .....	835
13.4.48 MasUpdGoodCtr .....	836
13.4.49 PhyUpd0GoodCtr .....	837
13.4.50 PhyUpd1GoodCtr .....	838
13.4.51 CtIUpd0GoodCtr .....	839
13.4.52 CtIUpd1GoodCtr .....	840
13.4.53 MasUpdFailCtr .....	841
13.4.54 PhyUpd0FailCtr .....	842
13.4.55 PhyUpd1FailCtr .....	843
13.4.56 PhyPerfCtrEnable .....	844
13.4.57 DfiHandshakeDelays1_pX (for X = 0; X <= 3) .....	846
13.4.58 DfiRespHandshakeDelays1_pX (for X = 0; X <= 3) .....	847
13.4.59 FspSkipList .....	848
13.4.60 PPGCReservedX (for X == 0) .....	849
13.4.61 PUBReservedP1_pX (PPGC) (for X = 0; X <= 3) .....	850
13.4.62 PhyInterruptOverride .....	851
13.4.63 PhyInterruptEnable .....	852
13.4.64 PhyInterruptFWControl .....	854

13.4.65 PhyInterruptMask .....	856
13.4.66 PhyInterruptClear .....	858
13.4.67 PhyInterruptStatus .....	860
13.4.68 ACSMRunCtrl .....	863
13.4.69 ACSMDone .....	865
13.4.70 ACSMStartAddr_pX (for X = 0; X <= 3) .....	866
13.4.71 ACSMStopAddr_pX (for X = 0; X <= 3) .....	867
13.4.72 ACSMLastAddr .....	868
13.4.73 ACSMAlgIncVal .....	869
13.4.74 ACSMAddressMask .....	870
13.4.75 ACSMOuterLoopRepeatCnt .....	871
13.4.76 ACSMCkeControl .....	872
13.4.77 ACSMCkeStatus .....	873
13.4.78 ACSMWckEnControl .....	874
13.4.79 ACSMWckEnStatus .....	875
13.4.80 ACSMRxEnPulse_pX (for X = 0; X <= 3) .....	876
13.4.81 ACSMRxValPulse_pX (for X = 0; X <= 3) .....	877
13.4.82 ACSMTxEnPulse_pX (for X = 0; X <= 3) .....	878
13.4.83 ACSMWrcsPulse_pX (for X = 0; X <= 3) .....	879
13.4.84 ACSMRdcsPulse_pX (for X = 0; X <= 3) .....	880
13.4.85 ACSMIInfiniteOLRC .....	881
13.4.86 ACSMDefaultAddr .....	882
13.4.87 ACSMDefaultCs .....	883
13.4.88 ACSMStaticCtrl .....	884
13.4.89 ACSMWckWriteStaticLoPulse_pX (for X = 0; X <= 3) .....	885
13.4.90 ACSMWckWriteStaticHiPulse_pX (for X = 0; X <= 3) .....	886
13.4.91 ACSMWckWriteTogglePulse_pX (for X = 0; X <= 3) .....	887
13.4.92 ACSMWckWriteFastTogglePulse_pX (for X = 0; X <= 3) .....	888
13.4.93 ACSMWckReadStaticLoPulse_pX (for X = 0; X <= 3) .....	889
13.4.94 ACSMWckReadStaticHiPulse_pX (for X = 0; X <= 3) .....	890
13.4.95 ACSMWckReadTogglePulse_pX (for X = 0; X <= 3) .....	891
13.4.96 ACSMWckReadFastTogglePulse_pX (for X = 0; X <= 3) .....	892
13.4.97 ACSMWckFreqSwStaticLoPulse_pX (for X = 0; X <= 3) .....	893
13.4.98 ACSMWckFreqSwStaticHiPulse_pX (for X = 0; X <= 3) .....	894
13.4.99 ACSMWckFreqSwTogglePulse_pX (for X = 0; X <= 3) .....	895
13.4.100 ACSMWckFreqSwFastTogglePulse_pX (for X = 0; X <= 3) .....	896
13.4.101 ACSMWckFreeRunMode_pX (for X = 0; X <= 3) .....	898
13.4.102 ACSMLowSpeedClockEnable .....	899
13.4.103 ACSMLowSpeedClockDelay .....	900
13.4.104 ACSMRptCntOverride_pX (for X = 0; X <= 3) .....	901
13.4.105 ACSMRptCntDbl_pX (for X = 0; X <= 3) .....	902
13.4.106 ACSMParityStatus .....	903
13.4.107 HwtLpCsEnBypass .....	904
13.4.108 ACSMNopAddr .....	905
13.4.109 SnoopCntrl .....	906
13.4.110 ACSMParityInvert .....	907
13.4.111 AcsmPsIdx .....	908
13.4.112 AcsmDynPtrCtrl .....	909
13.4.113 FspState .....	910

13.4.114 AcsmMapTableX (for X = 0; X <= 63)	911
13.4.115 AcsmStartAddrXlatValX (for X = 0; X <= 63)	912
13.4.116 AcsmStopAddrXlatValX (for X = 0; X <= 63)	913
<b>13.5 DWC_DDRPHYA_INITENGj_Pk Registers</b>	<b>914</b>
13.5.1 DVFSCEn_pX (for X = 0; X <= 3)	915
13.5.2 MtestMuxSel	916
13.5.3 StartVector0bX (for X = 0; X <= 15)	917
13.5.4 PhyInLPX (for X = 0; X <= 3)	918
13.5.5 INITENGParityInvert	919
13.5.6 ScratchPadINITENG	920
13.5.7 PieCtrlStartVec0_pX (for X = 0; X <= 3)	921
13.5.8 PIEProgErrStatus	922
13.5.9 PIEParityInvert	924
13.5.10 PIEParityStatus	925
13.5.11 Seq0BDDisableFlagX (for X = 0; X <= 31)	926
13.5.12 Seq0BCSRUpperWrData	927
13.5.13 Seq0BCSRUpperRdData	928
13.5.14 WaitCondUC	929
13.5.15 PhyMstrPMValOverride	930
13.5.16 PIEDebugStatus	931
13.5.17 DestPstate	932
13.5.18 DfiFreqXlatDestPStateX (for X = 0; X <= 15)	933
13.5.19 Seq0BGPR0_pX (for X = 0; X <= 3)	934
13.5.20 Seq0BGPR1_pX (for X = 0; X <= 3)	935
13.5.21 Seq0BGPR2_pX (for X = 0; X <= 3)	936
13.5.22 Seq0BGPR3_pX (for X = 0; X <= 3)	937
13.5.23 Seq0BGPR4_pX (for X = 0; X <= 3)	938
13.5.24 Seq0BGPR5_pX (for X = 0; X <= 3)	939
13.5.25 Seq0BGPR6_pX (for X = 0; X <= 3)	940
13.5.26 Seq0BGPR7_pX (for X = 0; X <= 3)	941
13.5.27 Seq0BGPR8_pX (for X = 0; X <= 3)	942
13.5.28 Seq0BGPR9_pX (for X = 0; X <= 3)	943
13.5.29 Seq0BGPR10_pX (for X = 0; X <= 3)	944
13.5.30 Seq0BGPR11_pX (for X = 0; X <= 3)	945
13.5.31 Seq0BGPR12_pX (for X = 0; X <= 3)	946
13.5.32 Seq0BGPR13_pX (for X = 0; X <= 3)	947
13.5.33 Seq0BGPR14_pX (for X = 0; X <= 3)	948
13.5.34 Seq0BGPR15_pX (for X = 0; X <= 3)	949
13.5.35 Seq0BGPR16_pX (for X = 0; X <= 3)	950
13.5.36 Seq0BGPR17_pX (for X = 0; X <= 3)	951
13.5.37 Seq0BGPR18_pX (for X = 0; X <= 3)	952
13.5.38 Seq0BGPR19_pX (for X = 0; X <= 3)	953
13.5.39 Seq0BGPR20_pX (for X = 0; X <= 3)	954
13.5.40 Seq0BGPR21_pX (for X = 0; X <= 3)	955
13.5.41 Seq0BGPR22_pX (for X = 0; X <= 3)	956
13.5.42 Seq0BGPR23_pX (for X = 0; X <= 3)	957
13.5.43 Seq0BGPR24_pX (for X = 0; X <= 3)	958
13.5.44 Seq0BGPR25_pX (for X = 0; X <= 3)	959
13.5.45 Seq0BGPR26_pX (for X = 0; X <= 3)	960

13.5.46 Seq0BGPR27_pX (for X = 0; X <= 3)	961
13.5.47 Seq0BGPR28_pX (for X = 0; X <= 3)	962
13.5.48 Seq0BGPR29_pX (for X = 0; X <= 3)	963
13.5.49 Seq0BGPR30_pX (for X = 0; X <= 3)	964
13.5.50 Seq0BGPR31_pX (for X = 0; X <= 3)	965
13.5.51 Seq0BGPRPage	966
13.5.52 Seq0BFixedAddrBits	967
13.5.53 PieDynFlagMaskX (for X = 0; X <= 3)	968
13.5.54 PieDynFlagSel	969
13.5.55 Seq0BPieGPRSel	970
13.5.56 Seq0BDLY0_pX (for X = 0; X <= 3)	971
13.5.57 Seq0BDLY1_pX (for X = 0; X <= 3)	972
13.5.58 Seq0BDLY2_pX (for X = 0; X <= 3)	973
13.5.59 Seq0BDLY3_pX (for X = 0; X <= 3)	974
13.5.60 Seq0BDLY4_pX (for X = 0; X <= 3)	975
13.5.61 Seq0BDLY5_pX (for X = 0; X <= 3)	976
13.5.62 Seq0BDLY6_pX (for X = 0; X <= 3)	977
13.5.63 Seq0BDLY7_pX (for X = 0; X <= 3)	978
13.5.64 Seq0BDLY8_pX (for X = 0; X <= 3)	979
13.5.65 Seq0BDLY9_pX (for X = 0; X <= 3)	980
13.5.66 Seq0BDLY10_pX (for X = 0; X <= 3)	981
13.5.67 Seq0BDLY11_pX (for X = 0; X <= 3)	982
13.5.68 Seq0BDLY12_pX (for X = 0; X <= 3)	983
13.5.69 Seq0BDLY13_pX (for X = 0; X <= 3)	984
13.5.70 Seq0BDLY14_pX (for X = 0; X <= 3)	985
13.5.71 Seq0BDLY15_pX (for X = 0; X <= 3)	986
13.5.72 Seq0BDLY16_pX (for X = 0; X <= 3)	987
13.5.73 Seq0BDLY17_pX (for X = 0; X <= 3)	988
13.5.74 Seq0BDLY18_pX (for X = 0; X <= 3)	989
13.5.75 Seq0BDLY19_pX (for X = 0; X <= 3)	990
13.5.76 Seq0BDLY20_pX (for X = 0; X <= 3)	991
13.5.77 Seq0BDLY21_pX (for X = 0; X <= 3)	992
13.5.78 Seq0BDLY22_pX (for X = 0; X <= 3)	993
13.5.79 Seq0BDLY23_pX (for X = 0; X <= 3)	994
13.5.80 Seq0BDLY24_pX (for X = 0; X <= 3)	995
13.5.81 Seq0BDLY25_pX (for X = 0; X <= 3)	996
13.5.82 Seq0BDLY26_pX (for X = 0; X <= 3)	997
13.5.83 Seq0BDLY27_pX (for X = 0; X <= 3)	998
13.5.84 Seq0BDLY28_pX (for X = 0; X <= 3)	999
13.5.85 Seq0BDLY29_pX (for X = 0; X <= 3)	1000
13.5.86 Seq0BDLY30_pX (for X = 0; X <= 3)	1001
13.5.87 Seq0BDLY31_pX (for X = 0; X <= 3)	1002
13.5.88 UpdDisFlagMask	1003
13.5.89 DisableFlagMask	1004
13.5.90 DisableFlagMasksX (for X = 0; X <= 1)	1005
13.5.91 RtrnMode_pX (for X = 0; X <= 3)	1006
13.5.92 RtrnModeMaskX (for X = 0; X <= 1)	1007
13.5.93 RtrnModeMask0sX (for X = 0; X <= 1)	1008
13.5.94 CtrlUpdAck	1009

13.5.95 Seq0BResetFixedAddrBits_pX (for X = 0; X <= 3) .....	1010
13.5.96 Seq0BCntrX (for X = 0; X <= 3) .....	1011
13.5.97 Seq0BCntr0Threshold_pX (for X = 0; X <= 3) .....	1012
13.5.98 Seq0BCntr1Threshold_pX (for X = 0; X <= 3) .....	1013
13.5.99 Seq0BCntr2Threshold_pX (for X = 0; X <= 3) .....	1014
13.5.100 Seq0BCntr3Threshold_pX (for X = 0; X <= 3) .....	1015
13.5.101 Seq0BCntrCtrl .....	1016
13.5.102 RtrnModeMask1sX (for X = 0; X <= 1) .....	1017
13.5.103 Seq0BGPR32_pX (for X = 0; X <= 3) .....	1018
13.5.104 Seq0BGPR33_pX (for X = 0; X <= 3) .....	1019
13.5.105 Seq0BGPR34_pX (for X = 0; X <= 3) .....	1020
13.5.106 Seq0BGPR35_pX (for X = 0; X <= 3) .....	1021
13.5.107 Seq0BGPR36_pX (for X = 0; X <= 3) .....	1022
13.5.108 Seq0BGPR37_pX (for X = 0; X <= 3) .....	1023
13.5.109 Seq0BGPR38_pX (for X = 0; X <= 3) .....	1024
13.5.110 Seq0BGPR39_pX (for X = 0; X <= 3) .....	1025
13.5.111 Seq0BGPR40_pX (for X = 0; X <= 3) .....	1026
13.5.112 Seq0BGPR41_pX (for X = 0; X <= 3) .....	1027
13.5.113 Seq0BGPR42_pX (for X = 0; X <= 3) .....	1028
13.5.114 Seq0BGPR43_pX (for X = 0; X <= 3) .....	1029
13.5.115 Seq0BGPR44_pX (for X = 0; X <= 3) .....	1030
13.5.116 Seq0BGPR45_pX (for X = 0; X <= 3) .....	1031
13.5.117 Seq0BGPR46_pX (for X = 0; X <= 3) .....	1032
13.5.118 Seq0BGPR47_pX (for X = 0; X <= 3) .....	1033
13.6 DWC_DDRPHYA_ZCALj_Pk Registers .....	1034
13.6.1 ZcalClkDiv .....	1035
13.6.2 ZCalClkInfo_pX (for X = 0; X <= 3) .....	1036
13.6.3 MtestMuxSel .....	1037
13.6.4 ZCALParityInvert .....	1038
13.6.5 ScratchPadZCAL .....	1039
13.6.6 ZCALReservedX (for X == 0) .....	1040
13.6.7 PUBReservedP1_pX (ZCAL) (for X = 0; X <= 3) .....	1041
13.6.8 ZCalBaseCtrl .....	1042
13.6.9 ZCalRate .....	1044
13.6.10 ZCalCtrl .....	1046
13.6.11 ZCalCompInvert .....	1048
13.6.12 ZCalCompOffset .....	1049
13.6.13 ZCalPUOffset .....	1050
13.6.14 ZCalCompOvr .....	1051
13.6.15 ZCalPUOvr .....	1052
13.6.16 ZCalClrFirstRunDone .....	1053
13.6.17 ZCalPDSearchGainTV .....	1054
13.6.18 ZCalAnaSettlingTime .....	1056
13.6.19 ZCalReset .....	1057
13.6.20 ZCalRun .....	1058
13.6.21 ZCalBusy .....	1059
13.6.22 ZCalCompResult .....	1060
13.6.23 ZCalPUResult .....	1061
13.6.24 ZCalPDResult .....	1062

13.6.25 ZCalCodePU .....	1063
13.6.26 ZCalCodePD .....	1064
13.6.27 ZCalCompSearchSeed .....	1065
13.6.28 ZCalPUSearchSeed .....	1066
13.6.29 ZCalPDSearchSeed .....	1067
13.6.30 ZCalCompSearchGainIV .....	1068
13.6.31 ZCalCompSearchGainTV .....	1070
13.6.32 ZCalPUSearchGainIV .....	1072
13.6.33 ZCalPUSearchGainTV .....	1074
13.6.34 ZCalPDSearchGainIV .....	1076
13.6.35 ZQUpdate .....	1078
13.6.36 ZQCalCodePUchX (for X = 0; X <= 1) .....	1079
13.6.37 ZQCalCodePDchX (for X = 0; X <= 1) .....	1080
13.6.38 ZQCalBaseCtrl .....	1081
13.6.39 ZQCalCodeOffsetPU .....	1082
13.6.40 ZQCalCodeOffsetPD .....	1083
13.6.41 ZQCalCodeOvrPU .....	1084
13.6.42 ZQCalCodeOvrPD .....	1085
13.6.43 ZQCalCodePUMax .....	1086
13.6.44 ZQCalCodePUMin .....	1087
13.6.45 ZQCalCodePDMAX .....	1088
13.6.46 ZQCalCodePDMIN .....	1089
13.6.47 ZCalStopClk_pX (for X = 0; X <= 3) .....	1090
13.7 DWC_DDRPHYA_DRTUBj Registers .....	1091
13.7.1 PieVecCfg .....	1092
13.7.2 PieInitVecSel .....	1093
13.7.3 DctShadowRegs .....	1094
13.7.4 DctWriteOnlyShadow .....	1095
13.7.5 UctWriteOnly .....	1096
13.7.6 UctWriteProt .....	1097
13.7.7 UctDatWriteOnly .....	1098
13.7.8 UctDatWriteProt .....	1099
13.7.9 UctlErr .....	1100
13.7.10 DRTUBParityInvert .....	1101
13.7.11 UCParityInvert .....	1102
13.7.12 ScratchPadDRTUB .....	1103
13.7.13 UcclkHclkEnables .....	1104
13.7.14 ArcEccIndications .....	1105
13.7.15 ArcIccmSbErrCtr .....	1106
13.7.16 ArcDccmSbErrCtr .....	1107
13.7.17 ArcSbCtrEnables .....	1108
13.7.18 ArcPmuEccCtl .....	1109
13.7.19 StartDCCMClear .....	1110
13.7.20 DCCMClearRunning .....	1111
13.7.21 PIEMicroReset .....	1112
13.7.22 CUSTPHYREV .....	1114
13.7.23 CUSTPUBREV .....	1115
13.7.24 PUBREV .....	1116
13.7.25 PUBVAR .....	1117

13.7.26 DfiFreqXlatX (for X = 0; X <= 15) .....	1118
<b>13.8 DWC_DDRPHYA_APBONLYj Registers .....</b>	<b>1119</b>
13.8.1 MicroContMuxSel .....	1120
13.8.2 ContextToMicro .....	1121
13.8.3 ExternalAHBReset .....	1122
13.8.4 TDRDisable .....	1123
13.8.5 UctShadowRegs .....	1124
13.8.6 BlockDfiShadowRegs .....	1125
13.8.7 CCMWriteBypassEnable .....	1126
13.8.8 DctWriteOnly .....	1127
13.8.9 DctWriteProt .....	1128
13.8.10 UctWriteOnlyShadow .....	1129
13.8.11 UctDatWriteOnlyShadow .....	1130
13.8.12 NeverGateCsrClock .....	1131
13.8.13 NeverGateAcCsrClock .....	1132
13.8.14 DfiCfgRdDataValidTicks .....	1133
13.8.15 DisableHMRdSpeedUp .....	1134
13.8.16 OverrideHMRdSpeedUp .....	1135
13.8.17 Dfi0DebugControl .....	1136
13.8.18 Dfi0DebugCaptureX (for X = 0; X <= 1) .....	1137
13.8.19 Dfi0DebugPerfCtrEn .....	1138
13.8.20 Dfi0DebugPerfCtr .....	1139
13.8.21 Dfi1DebugControl .....	1140
13.8.22 Dfi1DebugCaptureX (for X = 0; X <= 1) .....	1141
13.8.23 Dfi1DebugPerfCtrEn .....	1142
13.8.24 Dfi1DebugPerfCtr .....	1143
13.8.25 APBONLYParityInvert .....	1144
13.8.26 WaitCondAPB .....	1145
13.8.27 PIERdDataValidTicks .....	1146
13.8.28 UCRdDataValidTicks .....	1147
13.8.29 ScratchPadAPBONLY .....	1148
13.8.30 MicroReset .....	1149
13.8.31 MicroResetPIEEn .....	1151
13.8.32 ClearPIESTallToMicro .....	1152
13.8.33 PIEMicroStallDelay .....	1153
13.8.34 SequencerOverride .....	1154
13.8.35 DfiInitCompleteShadow .....	1155
13.8.36 APBONLYReservedX (for X == 0) .....	1156

# Revision History

---

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



- Note**
- Links and references to section, table, figure, and page numbers in this table are only assured to be valid for the version in which the change is made.
  - In some instances, documentation-only updates occur. The Synopsys IP product information (<https://www.synopsys.com/designware-ip.html>) has the latest documentation.
  - Documentation only updates are designated using the following numbering structure <phy\_version>\_d<x>.
-

PUB Version	Databook Date	Description
2.30a	May 7, 2024	<p>Updated:</p> <ul style="list-style-type: none"> <li>■ “System Components” on page 43</li> <li>■ “Highlights” on page 47</li> <li>■ “PHY AC Configuration” on page 54</li> <li>■ Table “Dual Channel AC Configuration” on page 55</li> <li>■ “Signal Descriptions” on page 73</li> <li>■ Table “Configurable Pipeline” on page 191</li> <li>■ “DfiClk” on page 143</li> <li>■ “PIIBypClk” on page 144</li> <li>■ “PIIRefClk” on page 144</li> <li>■ “APBCLK” on page 144</li> <li>■ “LP3/IO Retention Save Registers” on page 161</li> <li>■ “PHY State Changes and dfi_frequency Encoding” on page 172</li> <li>■ Table “dfi_frequency Encoding” on page 173</li> <li>■ Table “Frequency Change Sequence Breakdown” on page 175</li> <li>■ “DFI Status Interface Latencies” on page 177</li> <li>■ Figure “Standard Product System Configuration: PLL bypass” on page 187</li> <li>■ “Compile-time PIPE Overview” on page 189</li> <li>■ “Interrupt Implementation” on page 206</li> <li>■ “Drift Tracking by MRR Snooping” on page 207</li> <li>■ Table “Comparison Between PPT (Retrain Only/PMI) and MRR Snoop Retraining” on page 209</li> <li>■ “PHY Configuration” on page 244</li> <li>■ Table “PHY_TOP Clocking During Scan” on page 253</li> <li>■ “DTO Loopback” on page 256</li> <li>■ “Digital Test Observation Pin” on page 260</li> <li>■ Table “Primary Digital Observation Pin Signals Available in Each PUB Dx Instance for Mux B” on page 268</li> <li>■ “Analog Observability” on page 273</li> <li>■ “Bypass Mode” on page 278</li> <li>■ “Bypass Mode Rules” on page 280</li> <li>■ “VIH/VIL Test Sequence” on page 290</li> <li>■ “DQ, DMI, CA, CK Test Sequence” on page 290</li> <li>■ “VOH/VOL Testing” on page 294</li> <li>■ “Register Descriptions” on page 349</li> </ul> <p>Added:</p> <ul style="list-style-type: none"> <li>■ “Controlled Power off Considerations” on page 150</li> <li>■ “ARC SRAM Write Mask Mapping” on page 215</li> <li>■ “Periodic Phase Training 2 (PPT2)” on page 237</li> <li>■ “Incremental Retraining Used in PPT2” on page 239</li> <li>■ “Automotive Support” on page 337</li> </ul>

PUB Version	Databook Date	Description
2.10_d1	March 15, 2024	<p>Updated:</p> <ul style="list-style-type: none"> <li>■ Instances of “DesignWare Cores” to “Synopsys IP” to align with corporate naming conventions</li> <li>■ “Signal Descriptions” on page 77</li> <li>■ “Register Descriptions” on page 357</li> </ul>
2.10a	October 5, 2023	<p>Updated:</p> <ul style="list-style-type: none"> <li>■ “Highlights” on page 47</li> <li>■ “Signal Descriptions” on page 73</li> <li>■ Table “DFI Write Timing Parameters” on page 129</li> <li>■ Table “WCK Control Timing Parameter” on page 131</li> <li>■ “DFI Update Timing Parameters” on page 135</li> <li>■ “Register Restore Timing” on page 167</li> <li>■ “DFI Status Interface Latencies” on page 177</li> <li>■ Table “Max tinit_complete Latency for LPDDR5” on page 183</li> <li>■ “Compile-time PIPE Overview” on page 189</li> <li>■ “Configurable PIPE Overview” on page 191</li> <li>■ “APB Interface Access Rules” on page 193</li> </ul> <p>(continues in the following page)</p>

PUB Version	Databook Date	Description
2.10a Continuation	October 5, 2023 Continuation	<p>Updated (Continued):</p> <ul style="list-style-type: none"> <li>■ “Drift Tracking by MRR Snooping” on page 207</li> <li>■ “PCLK Duty Cycle Detection” on page 219</li> <li>■ “Impedance Calibration” on page 227</li> <li>■ “PHY Core DVFS Support” on page 235</li> <li>■ “Scan Mode” on page 250</li> <li>■ “DTO Loopback” on page 256</li> <li>■ Table “Primary Digital Observation Pin Signals Available in Each DQS Pair for Mux A” on page 267</li> <li>■ Table “Primary Digital Observation Pin Signals Available in Each DQS Pair for Mux B” on page 268</li> <li>■ Table “Primary Digital Observation Pin Signals Available in Each PUB AC Instance” on page 266</li> <li>■ “Bypass Mode Rules” on page 280</li> <li>■ “Bypass Mode Architecture” on page 282</li> <li>■ Figure “Bypass/Flyover Controls 1” on page 282</li> <li>■ Figure “Bypass/Flyover Controls 2” on page 283</li> <li>■ Figure “Bypass/Flyover Controls 3” on page 284</li> <li>■ “VIH/VIL Testing” on page 287</li> <li>■ Figure “Bypass Path in the PHY During VIH/VIL Testing” on page 297</li> <li>■ “VIH/VIL Test Sequence” on page 290</li> <li>■ “DQ, DMI, CA, CK Test Sequence” on page 290</li> <li>■ “DQS, WCK Test Sequence” on page 292</li> <li>■ “PclkPtrInitVal Impacts From Construction Skews” on page 305</li> <li>■ Table “PclkPtrInitVal Pointer Separation Ranges” on page 305</li> <li>■ “LPDDR4X Mode CDD” on page 307</li> <li>■ “LPDDR5 Mode CDD” on page 309</li> <li>■ Table “PHY Skew Limits [LPDDR4X]” on page 313</li> <li>■ “DQ-to-DQ Arrival Time Mismatch” on page 315</li> <li>■ Table “Power Saving Modes when Dfi_lp_data_req Asserts” on page 320</li> <li>■ “Register Descriptions” on page 349</li> </ul> <p>Added:</p> <ul style="list-style-type: none"> <li>■ “DRAM ZQ Calibration” on page 223</li> <li>■ “Per-pin DFE Support” on page 223</li> <li>■ “Automotive Support” on page HIDDEN</li> </ul> <p>Removed:</p> <ul style="list-style-type: none"> <li>■ Section “LPDDR5 Write Clock (WCK)” from “LPDDR5X/5 Specific Features” on page 220</li> </ul>

PUB Version	Databook Date	Description
1.21a	June 16, 2023	<p>Updated:</p> <ul style="list-style-type: none"> <li>■ Table “Dual Channel AC Configuration” on page 55</li> <li>■ Table “DFI Control Timing Parameters” on page 128</li> <li>■ Table “WCK Control Timing Parameter” on page 131</li> <li>■ “(J) Initialize the PHY to Mission Mode through DFI Initialization” on page 154</li> <li>■ “LP3/IO Retention Save Registers” on page 161</li> <li>■ “LP3 / IO Retention Enter” on page 166</li> <li>■ “LP3 / IO Retention Exit” on page 164</li> <li>■ “DFI Status Interface Latencies” on page 181</li> <li>■ “ARC-HS Features” on page 214</li> <li>■ “ICCM and DCCM Program Loading” on page 215</li> <li>■ “Write X Function Support” on page 223</li> <li>■ “Link ECC Support” on page 220</li> <li>■ “Starting Calibration” on page 231</li> <li>■ “Customer Defines” on page 245</li> <li>■ “ASST Overview” on page 251</li> <li>■ “Bypass Mode Architecture” on page 293</li> <li>■ “Selecting VREF Source” on page 300</li> <li>■ “LPDDR4X Mode CDD” on page 307</li> <li>■ “LPDDR5 Mode CDD” on page 309</li> <li>■ Table “PHY Skew Limits [LPDDR5X/5]” on page 312</li> <li>■ Table “PHY Skew Limits [LPDDR4X]” on page 327</li> <li>■ “DQ-to-DQ Arrival Time Mismatch” on page 315</li> <li>■ “Register Descriptions” on page 349</li> </ul> <p>Added:</p> <ul style="list-style-type: none"> <li>■ LPDDR5 Write Clock (WCK)</li> <li>■ “DRAM ZQ Calibration” on page 226</li> </ul> <p>Removed:</p> <ul style="list-style-type: none"> <li>■ Table “PllAnaTstSel Test Bit Decoder” from section “Analog Observability” on page 273</li> </ul>

PUB Version	Databook Date	Description
1.20a	March 21, 2023	<p>Updated:</p> <ul style="list-style-type: none"> <li>■ “LPDDR5/5x-Specific Features” on page 50</li> <li>■ Table “DBYTE Configuration (DWC_LPDDR5XPHY_DBYTE_DM_ENABLED is defined)” on page 56</li> <li>■ “Signal Descriptions” on page 73</li> <li>■ Table “DFI Control Timing Parameters” on page 128</li> <li>■ “Reset” on page 148</li> <li>■ “LP3/IO Retention Save Registers” on page 161</li> <li>■ “LP3 / IO Retention Exit” on page 164</li> <li>■ Table “Frequency Change Sequence Breakdown” on page 175</li> <li>■ “Clocking Architecture and Configurations” on page 186</li> <li>■ “LPDDR5X/5 Specific Features” on page 220</li> <li>■ “ARC-HS Microcontroller” on page 211</li> <li>■ “Starting Calibration” on page 231</li> <li>■ “DRAM DVFSQ Support” on page 235</li> <li>■ “ASST Overview” on page 251</li> <li>■ “VOH/VOL Testing” on page 294</li> <li>■ “VOH/VOL Test Sequence” on page 297</li> <li>■ “ODT Impedance Test Sequence” on page 299</li> <li>■ “Rank-to-Rank Spacing” on page 306</li> <li>■ Table “PHY Skew Limits [LPDDR5X/5]” on page 312</li> <li>■ Table “Power Saving Modes when Dfi_lp_ctrl_req Asserts” on page 322</li> <li>■ “Multi-channel Operation” on page 330</li> <li>■ “Register Descriptions” on page 349</li> </ul>
1.10a_d2	October 27, 2022	<p>Updated:</p> <ul style="list-style-type: none"> <li>■ “DFI Frequency Change” on page 171</li> <li>■ “PHY Core DVFS Support” on page 235</li> <li>■ “Design for Test” on page 249</li> </ul>
1.10a_d1	October 12, 2022	<p>Updated:</p> <ul style="list-style-type: none"> <li>■ Table “DFI Write Parameters” on page 129</li> <li>■ “APB Interface Access Rules” on page 193</li> <li>■ “ARC and SRAM Clocking” on page 216</li> <li>■ Table “Primary Digital Observation Pin Signals Available in Each PUB AC Instance” on page 266</li> <li>■ “Rank-to-Rank Spacing” on page 306</li> <li>■ “SRAM” on page 200</li> <li>■ “PIE Instruction SRAM” on page 204</li> <li>■ “ASST Overview” on page 251</li> <li>■ “PclkPtrInitVal Impacts From Construction Skews” on page 305</li> <li>■ “ICCM and DCCM Program Loading” on page 215</li> <li>■ “ARC-HS Microcontroller” on page 211</li> <li>■ “PHY Latency” on page 304</li> </ul>

PUB Version	Databook Date	Description
1.10a	September 27, 2022	<p>Updated:</p> <ul style="list-style-type: none"><li>■ PUB version and date</li><li>■ “Register Descriptions” on page 349</li></ul>

PUB Version	Databook Date	Description
1.00a	September 2, 2022	<p>Updated:</p> <ul style="list-style-type: none"> <li>■ “Preface” on page 37</li> <li>■ “PUB Area” on page 57</li> <li>■ “PUB Utilization” on page 58</li> <li>■ “Definition of Terms and Acronyms” on page 61</li> <li>■ “Functional Overview” on page 65</li> <li>■ “Signal Descriptions” on page 73</li> <li>■ “Resets” on page 146</li> <li>■ Table “dfi_frequency Encoding” on page 173</li> <li>■ Figure “PHY Transition Diagrams” on page 174</li> <li>■ Table “Frequency Change Sequence Breakdown” on page 175</li> <li>■ “DRAM State During Frequency and Lower Power Changes” on page 175</li> <li>■ “DFI Status Interface Latencies” on page 177</li> <li>■ “APB” on page 193</li> <li>■ “APB Interface Access Rules” on page 193</li> <li>■ “Drift Tracking by MRR Snooping” on page 207</li> <li>■ “External AHB-Lite Interface” on page 217</li> <li>■ “PCLK Duty Cycle Detection” on page 219</li> <li>■ “Strobe-less Read Mode” on page 221</li> <li>■ “Scan Mode” on page 250</li> <li>■ “ASST Overview” on page 251</li> <li>■ “PHY_TOP Scan Clocks” on page 253</li> <li>■ “Loopback BIST (ATE) Firmware Setup and Execution” on page 253</li> <li>■ “Delay Element Testing” on page 256</li> <li>■ “Delay Test Hardware” on page 257</li> <li>■ “Digital Test Observation Pin” on page 260</li> <li>■ “Using Two Observation Pins to Assist PLL Debug” on page 271</li> <li>■ “Bypass Mode” on page 278</li> <li>■ “Bypass Mode Architecture” on page 282</li> <li>■ Table “Default Attenuation Level” on page 302</li> <li>■ “VOH/VOL Test Sequence” on page 297</li> <li>■ “ODT Impedance Testing” on page 298</li> <li>■ “PHY Skew Limits” on page 312</li> <li>■ “DFI Master Interface” on page 328</li> <li>■ “Supported DFI Sideband Interfaces” on page 329</li> <li>■ “Register Descriptions” on page 349</li> <li>■ Naming harmonization across the PUB databook</li> </ul> <p>Added:</p> <ul style="list-style-type: none"> <li>■ “PHY Core DVFS Support” on page 235</li> </ul> <p>Removed:</p> <ul style="list-style-type: none"> <li>■ “Automotive Support” from chapter “Architecture” on page 185</li> </ul>

PUB Version	Databook Date	Description
0.90a	June 1, 2022	<p>Updated:</p> <ul style="list-style-type: none"> <li>■ <a href="#">“Compatible Standards” on page 53</a></li> <li>■ <a href="#">“Limitation and Restrictions” on page 60</a></li> <li>■ <a href="#">“Dual Channel AC Configuration” on page 55</a></li> <li>■ <a href="#">“PUB Area” on page 57</a></li> <li>■ <a href="#">“DFI Control Timing Parameters” on page 128</a></li> <li>■ <a href="#">“DFI PHY MASTER Timing Parameters” on page 138</a></li> <li>■ <a href="#">“PHY Fast Standby (LP2)” on page 159</a></li> <li>■ <a href="#">“LP3 / IO Retention Exit” on page 164</a></li> <li>■ <a href="#">“Register Restore Timing” on page 167</a></li> <li>■ <a href="#">“PHY State Changes and dfi_frequency Encoding” on page 172</a></li> <li>■ <a href="#">“DFI Status Interface Latencies” on page 177</a></li> <li>■ <a href="#">“APB Interface Access Rules” on page 193</a></li> <li>■ <a href="#">“SRAM” on page 200</a></li> <li>■ <a href="#">“PIE” on page 201</a></li> <li>■ <a href="#">“Notes about ECC support” on page 216</a></li> <li>■ <a href="#">“Digital Observability” on page 258</a></li> <li>■ <a href="#">“VOH/VOL Testing” on page 294</a></li> <li>■ <a href="#">“ODT Impedance Testing” on page 298</a></li> <li>■ <a href="#">“LPDDR4X Mode CDD” on page 307</a></li> <li>■ <a href="#">“Specifying Register Addresses” on page 346</a></li> </ul>

PUB Version	Databook Date	Description
0.90a_pre1	April 18, 2022	<p>Updated:</p> <ul style="list-style-type: none"> <li>■ “System Overview” on page 42</li> <li>■ “Highlights” on page 47</li> <li>■ “I/O Features” on page 52</li> <li>■ Figure “Functional Overview” on page 67</li> <li>■ Figure “Single Channel LPDDR DFI Connection” on page 69</li> <li>■ Figure “Dual Channel LPDDR DFI Connection” on page 70</li> <li>■ Table “DFI Write Timing Parameters” on page 129</li> <li>■ Table “DFI Read Data Timing Parameters” on page 134</li> <li>■ Table “DFI PHY MASTER Timing Parameters” on page 138</li> <li>■ Figure “Cold Reset Operation” on page 148</li> <li>■ “Reset” on page 148</li> <li>■ “PRESETn_APB” on page 149</li> <li>■ “WRSTN” on page 149</li> <li>■ Figure “Example Timing Diagram for PHY Initialization” on page 155</li> <li>■ Table “PHY Low Power State Modes” on page 157</li> <li>■ “LP3 / IO Retention Enter” on page 162</li> <li>■ “LP3 / IO Retention Exit” on page 164</li> <li>■ “DFI Status Interface” on page 169</li> <li>■ “DFI Initialization” on page 169</li> <li>■ “State of DRAM on first initialization” on page 170</li> <li>■ “DFI Frequency Change” on page 171</li> <li>■ Table “dfi_frequency Encoding” on page 173</li> <li>■ Figure “PHY Transition Diagrams” on page 174</li> <li>■ “DFI Status Interface Latencies” on page 177</li> <li>■ “APB Interface Access Rules” on page 193</li> <li>■ “SRAM” on page 200</li> <li>■ “ARC-HS Microcontroller” on page 211</li> <li>■ “Impedance Calibration” on page 227</li> <li>■ “Starting Calibration” on page 231</li> <li>■ “DRAM DVFSQ Support” on page 235</li> <li>■ “Automotive Support” on page HIDDEN</li> <li>■ “Customer Defines” on page 245</li> <li>■ “Scan Mode” on page 250</li> <li>■ “ASST Overview” on page 251</li> <li>■ Table “PHY_TOP Clocking During Scan” on page 253</li> <li>■ “Digital Test Observation Pin” on page 260</li> </ul> <p>(continues in the following page)</p>

PUB Version	Databook Date	Description
0.90a_pre1	April 18, 2022	<ul style="list-style-type: none"><li>■ Table “PLL Digital Test Outputs to Bump Mapping” on page 272</li><li>■ Table “ODTSeg120 CSR Settings” on page 297</li><li>■ Table “ODTSeg120 CSR Settings” on page 297</li><li>■ “Quiescent Current (IDQ) Measurement” on page 301</li><li>■ “Input Pin Leakage Measurement” on page 302</li><li>■ Table “PclkPtrInitVal Pointer Separation Ranges” on page 305</li><li>■ “PHY Skew Limits” on page 312</li><li>■ Table “Power Saving Modes when Dfi_Lp_data_req Asserts” on page 320</li><li>■ Table “DFI LP Control Interface” on page 322</li></ul>

PUB Version	Databook Date	Description
0.75a	December 23, 2021	<p>Updated:</p> <ul style="list-style-type: none"> <li>■ “<a href="#">Highlights</a>” on page <a href="#">47</a></li> <li>■ “<a href="#">LPDDR5/5x-Specific Features</a>” on page <a href="#">50</a></li> <li>■ “<a href="#">I/O Features</a>” on page <a href="#">52</a></li> <li>■ “<a href="#">Compatible Standards</a>” on page <a href="#">53</a></li> <li>■ “<a href="#">Single Channel AC Configuration</a>” on page <a href="#">54</a></li> <li>■ “<a href="#">PUB Utilization</a>” on page <a href="#">58</a></li> <li>■ “<a href="#">Limitation and Restrictions</a>” on page <a href="#">60</a></li> <li>■ “<a href="#">Signal Descriptions</a>” on page <a href="#">73</a></li> <li>■ “<a href="#">DFI Control Timing Parameters</a>” on page <a href="#">128</a></li> <li>■ “<a href="#">DFI WCK Control Timing Parameters (LPDDR5/5X)</a>” on page <a href="#">131</a></li> <li>■ “<a href="#">DFI Read Data Timing Parameters</a>” on page <a href="#">134</a></li> <li>■ “<a href="#">Warm Reset Operation</a>” on page <a href="#">147</a></li> <li>■ “<a href="#">BP_PWROK</a>” on page <a href="#">150</a></li> <li>■ “<a href="#">Cold Reset Operation</a>” on page <a href="#">148</a></li> <li>■ “<a href="#">Example Timing Diagram for PHY Initialization</a>” on page <a href="#">155</a></li> <li>■ “<a href="#">PHY Low Power State Modes</a>” on page <a href="#">157</a></li> <li>■ “<a href="#">LP3/IO Retention (LP3)</a>” on page <a href="#">160</a></li> <li>■ “<a href="#">LP3 IO Retention Exit</a>” on page <a href="#">167</a></li> <li>■ “<a href="#">PHY Transition Diagrams</a>” on page <a href="#">174</a></li> <li>■ “<a href="#">Frequency Change Sequence Breakdown</a>” on page <a href="#">175</a></li> <li>■ “<a href="#">DRAM State During DFI Status Update Requests</a>” on page <a href="#">176</a></li> <li>■ “<a href="#">Compile-time PIPE Overview</a>” on page <a href="#">189</a></li> <li>■ “<a href="#">Configurable Pipeline</a>” on page <a href="#">191</a></li> <li>■ “<a href="#">TDR Procedure for CSR Read</a>” on page <a href="#">196</a></li> <li>■ “<a href="#">SRAM</a>” on page <a href="#">200</a></li> <li>■ “<a href="#">Drift Tracking by MRR Snooping</a>” on page <a href="#">207</a></li> <li>■ “<a href="#">ARC-HS Microcontroller</a>” on page <a href="#">211</a></li> <li>■ “<a href="#">Supported Configuration</a>” on page <a href="#">245</a></li> <li>■ “<a href="#">Scan Mode</a>” on page <a href="#">250</a></li> <li>■ “<a href="#">Scan Testing Sequence</a>” on page <a href="#">252</a></li> <li>■ “<a href="#">DQSX1 Loopback</a>” on page <a href="#">255</a></li> <li>■ “<a href="#">ACX1/CSX1 Loopback</a>” on page <a href="#">255</a></li> <li>■ “<a href="#">CKX2 Loopback</a>” on page <a href="#">256</a></li> <li>■ “<a href="#">Overall Test Selection Structure</a>” on page <a href="#">259</a></li> </ul> <p>(continues in the following page)</p>

PUB Version	Databook Date	Description
0.75a	December 23, 2021	<ul style="list-style-type: none"> <li>■ “Digital Test Observation Pin” on page 260</li> <li>■ “MtestMuxSel PUB or Hard Macro Selection” on page 262</li> <li>■ “BP_ATO in ZCAL Hard Macro” on page 274</li> <li>■ “Bypass Mapping from the Bypass Signal to the Bumps (Single Channel)” on page 278</li> <li>■ “BypassMode Control with SCAN_Mode” on page 282</li> <li>■ “Bypass/Flyover Controls 2” on page 283</li> <li>■ “Selecting VREF Source” on page 289</li> <li>■ “VIH/VIL Test Sequence” on page 290</li> <li>■ “DQ, DMI, CA, CK Test Sequence” on page 290</li> <li>■ “VOH/VOL Testing” on page 294</li> <li>■ “VOH/VOL Testing” on page 296</li> <li>■ “VOH/VOL Test Sequence” on page 297</li> <li>■ “ODT Impedance Test Sequence” on page 299</li> <li>■ “Quiescent Current (IDQ) Measurement” on page 301</li> <li>■ “PHY Latency” on page 304</li> <li>■ “PclkPtrInitVal Pointer Separation Ranges” on page 305</li> <li>■ “Preamble and Postamble Restrictions” on page 311</li> <li>■ “Collisions Between PHY Update and DFI LP” on page 325</li> <li>■ “Register Descriptions” on page 349</li> </ul> <p>Added:</p> <ul style="list-style-type: none"> <li>■ “WDQS Extension” on page 225</li> <li>■ “Bypass Mapping from the Bypass Signal to the Bumps (Dual Channel)” on page 279</li> </ul>
0.50a	September 6, 2021	<ul style="list-style-type: none"> <li>■ Initial release</li> </ul>



# Preface

---

This databook describes the DWC LPDDR5X/5/4X PHY Utility Block (PUB):

## Databook Organization

The chapters of this databook are organized as follows:

- [Chapter 1, “Product Overview”](#) provides an overview of the DWC LPDDR5X/5/4X PHY, and key features
- [Chapter 2, “Functional Overview”](#) provides a functional description and signal description.
- [Chapter 3, “Signal Descriptions”](#) provides signal descriptions.
- [Chapter 4, “Signal Mapping”](#) provides signal mapping information.
- [Chapter 5, “Parameter Description”](#) provides design parameters descriptions.
- [Chapter 6, “Clocks, Resets, Power Domains, Initialization, and Power Management”](#) provides clocking, reset, initialization and power management of the DWC LPDDR5X/5/4X PHY PUB.
- [Chapter 7, “Architecture”](#) describes the architecture of the DWC LPDDR5X/5/4X PHY PUB.
- [Chapter 8, “System Integration”](#) provides information for integrating the DDR PHY with either the Synopsys IP Protocol or Memory Controllers or a non-Synopsys memory controller.
- [Chapter 9, “Design for Test”](#) provides information about the boundary scan register, including its signals and cell structures.
- [Chapter 10, “Timing”](#) describes all the detailed timing for the interface between the controller and the DWC LPDDR5X/5/4X PHY.
- [Chapter 11, “Automotive Support”](#) provides an overview regarding Automotive support.
- [Chapter 12, “Register Overview”](#) provides an overview of the memory map of the DWC LPDDR5X/5/4X PHY and the PHYCTRL block of the DWC LPDDR5X/5/4X PHY.
- [Chapter 13, “Register Descriptions”](#) provides the memory map and register description of the DWC LPDDR5X/5/4X PHY and the PHYCTRL block of the DWC LPDDR5X/5/4X PHY.

## Reference Documentation

- [Synopsys PHY IP LPDDR5X/5/4X PHY Databook, Synopsys, Inc.](#)
- [Synopsys PHY IP LPDDR5X/5/4X PHY Databook Utility Block \(DWC LPDDR5X/5/4X PHY\) Release Notes, Synopsys, Inc.](#)

## STAR on the Web (SotW)

You must review all STARs on the Web (SotWs) associated with your product. SotWs are considered a part of the Synopsys documentation suite, and show critical information related to your product. To review product SotWs, refer to the Synopsys IP product information:

<https://www.synopsys.com/designware-ip.html>

## Synopsys Statement on Inclusivity and Diversity

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

## Web Resources

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

## Customer Support

For customer support, Synopsys provides various methods for contacting Customer Support, as follows:

- For *fastest response*, enter a case through SolvNetPlus:
  - a. <https://solvnetplus.synopsys.com>



**Note** SolvNetPlus does not support Internet Explorer.

- b. Click the **Cases** menu and then click **Create a New Case** (below the list of cases).
- c. Complete the mandatory fields that are marked with an asterisk and click **Save**.

Make sure you include the following:

- **Product L1:** Designware Cores
- **Product L2:** LPDDR5X/5/4X PHY
- **Version:** 2.30a

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

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

- Or, send an e-mail message to [support\\_center@synopsys.com](mailto:support_center@synopsys.com) (your e-mail will be queued and then, on a first-come, first-served basis, manually routed to the correct support engineer):
  - Include the Product L1 and Product L2 names, process, and Version number in your e-mail so it can be routed correctly.
  - For simulation issues, include the timestamp of any signals or locations in waveforms that are not understood
- Or, telephone your local support center:
  - North America:  
Call 1-800-245-8005 from 7 AM to 5:30 PM Pacific time, Monday through Friday.
  - All other countries:  
<https://www.synopsys.com/support/global-support-centers.html>



# Product Overview

---

The following sections are included in this chapter:

- “System Overview” on page 42
- “System Components” on page 43
- “Firmware/PHYINIT Usage Model” on page 45
- “Memory Controller Solution” on page 46
- “LPDDR5X/5/4X PHY Features” on page 47
- “Compatible Standards” on page 53
- “Supported System Configurations” on page 54
- “PUB Area” on page 57
- “PUB Utilization” on page 58
- “Timing Closure” on page 59
- “Limitation and Restrictions” on page 60
- “Definition of Terms and Acronyms” on page 61

## 1.1 System Overview

The LPDDR5X/5/4X PHY interface solution consists of the Soft IP PHY Utility Block (PUB\_TOP) and the PHY hard macro IPs.

The Soft IP provides control features to ease the customer implementation of digitally controlled features of the PHY such as initialization, remap logic, training, VT drift compensation, built-in self-test features to provide support for production testing, and provides a DFI interface to the PHY.

The LPDDR5X/5/4X PHY has configuration and status registers (CSRs) that are accessible via a configuration port. The configuration port is an APB interface. There is also a separate TDR interface to improve ease of access for test functions.

A complete memory interface solution consists of the PHY and Memory controller as shown in [Figure 2-1](#) on page [67](#).

## 1.2 System Components

The full DWC LPDDR5X/5/4X PHY IP solution consists of multiple components, including a combination of the following hard IP, soft IP and Firmware:

- PHY\_TOP consists of multiple components, including a combination of the following soft IP.
- The PUB\_TOP is a soft IP component to be used with the DWC PHY. Features include
- PUB\_ACX/PUB\_DX : Converts DFI signals into PHY internal control sequences to write data, read data, addr/ cmd to interface DRAM.
- PUB\_ZCAL : Impedance calibration and IO VT compensation.
- PUB\_PAC : PClk Synchronization and Data Path initialization.
- TUB provides control features to ease the implementation of digitally controlled functions of the PHY such as initialization, DRAM interface training, delay line calibration and VT compensation, and programmable configuration controls. It provides a DFI interface and configuration port to access PHY configuration resistors. The configuration port is compatible with an APB interface.

The following section describes Hard Macro (HM).

**DDDRPHY DBYTE:** The DBYTE is the data receive (RX)/ transmit (TX) building component for the DDR PHY. DBYTE can be constructed using HM components DX4/DX5. Features of DBYTE include

- High speed digital logic pipeline for transmitting data paths to the SDRAM
- High speed digital logic pipeline for receiving data paths from the SDRAM
- Drivers and receivers for the DQ and DQS signals as well as all the necessary high voltage analog macros to interface off chip
- I/O components that include PVT-compensated on-die termination (ODT) and output impedance drivers and receivers

**DDDRPHY ACX:** The ACX is the PHY component for the address/command lanes. ACX can be constructed using HM components ACX2 and CKX2. Features of the ACX include:

- High-speed digital logic pipeline for transmitting address/command to the SDRAM
- High speed digital logic pipeline for receiving data paths for loopback mode
- I/O components that include PVT-compensated output impedance drivers and receivers for loopback mode
- PHY Digital Test Out debug port (BP.DTO)
- Related utility functions to provide control and configuration

**DDDRPHY CMOS:** Features of CMOS include:

- Digital logic pipeline for transmitting MEMRESET\_L to the SDRAM
- Power on reset (POR) circuit (BP\_PWROK)

**DDDRPHY PAC:** The PAC hard macro is the location of shared resources in the PHY. Features of the PAC hard macro include:

- The PLL block generates the DRAM bit-rate clock for data transmit and high speed digital pipelines
- Analog Observability port for PLL (BP\_ATO\_PLL)

**DDDRPHY ZCAL:** The ZCAL is the location for transmitter impedance calibration. ZCAL is constructed using HM components ZCAL. Features of the ZCAL include:

- Transmitter replica for PU and PD impedance calibration
- Comparator circuit for impedance calibration (ZCALANA)
- Related utility functions to provide control and configuration
- Analog Observability port (BP\_ATO)

ATE Firmware for screening and characterizing the PHY during manufacturing.

Training Firmware for optimizing the channel between the PHY and the DRAM devices.

PHYINIT utility for creating the configuration and initialization sequence for the PHY.

## 1.3 Firmware/PHYINIT Usage Model

Firmware and PHYINIT are often updated as improvements are made based on JEDEC standard updates, training algorithms, etc. Using Firmware/PHYINIT in different usage scenarios than below will compromise a customer's ability to use the latest versions and/or take bug fixes.

### 1.3.1 ATE Firmware Usage Model

Synopsys expects customers to use ATE Firmware to screen parts during manufacturing, for parametric characterization and to aid in silicon debug. Synopsys expects customers to be able to take patches or updates.

### 1.3.2 Training Firmware Usage Model

Synopsys expects customers to use Training Firmware to train the PHY before entering mission mode and to aid in silicon debug. Synopsys expects customers to be able to take patches or updates.

### 1.3.3 PHYINIT Usage Model

Synopsys expects customers to use the PHYINIT utility to create the register writes needed to initialize the PHY and memory, including loading and executing the training firmware. Synopsys expects customers to be able to take patches or updates.

PHYINIT is intended for use of direct licensees of the DDR PHY to create register write sequences. Only this output should be shared with end customers with advanced written permission from Synopsys.

## 1.4 Memory Controller Solution

A memory controller solution is required for PHY applications. The solution uses the PUB's DFI interface to connect between the PHY and the memory controller.

## 1.5 LPDDR5X/5/4X PHY Features

### 1.5.1 Highlights



**Note** Data rates are hard-macro dependent check the respective PHY Databook for maximum data rate support.

- High-performance DDR PHY
- Low latency, small area, low power
- Supports following JEDEC standards:
  - LPDDR5X-9600
  - LPDDR5-6400
  - LPDDR4X-4267
- DFI 5.0-compliant controller interface
- Standard DFI 1:2 and 1:4 mode support (selected at boot time)
  - LPDDR5/5X supports DFI 1:2 mode from 40 Mbps to 3200 Mbps.
  - LPDDR5/5X supports DFI 1:4 mode from 40 Mbps to 9600 Mbps.
  - LPDDR4X supports DFI 1:2 as well as DFI 1:4 mode from 20 Mbps to 4267 Mbps
- DFI 1:2 / 1:4 FSP-switchable mode supported
- Optional dual channel DFI supported
- Optional pipeline support to close timing at DFI interface
- Independent per-channel control for DFI\_LP / DFI\_PHYUPDATE / DFI\_CTRLUPDATE
- 2-rank LPDDR5 or LPDDR5X support with 32 DQ bits, 4 DMI bits, 4 WCK strobe pairs, 4 RDQS strobe pairs, 2 CA channels, 2 CK pairs and 1 MEMRESET

#### Flexible channel architecture

- Support for 1 or 2 independent LPDDR5/5X channels
- Supports 16-bit data path widths per channel
  - For 32-bit data path, two options supported:
    - One instance of dual-channel PHY
    - Two instances of single channel PHY
  - For 64-bit data path, two options supported:
    - Two instances of dual-channel PHY
    - Four instances of single channel PHY
  - When multiple PHY instances connect to a single SDRAM package, shared signals require special handling.
    - Refer to Training Firmware Application Note for details.
- Supports up to 4 trained states/frequencies with low switching time

- Designed for rapid integration with Synopsys memory or protocol controllers for a complete DDR interface solution
  - Supports fast FSP switch for LPDDR5/5X/4X between 1:2 and 1:4 mode without the need to change DfiClk or lock the PLL again
- All signaling and DRAM controls registers are optimized for per-pstate power modes
- VT compensated delay lines for DQS centering, read/write leveling, and per bit deskew
- PHY supports many DRAM packaging options, including:
  - SDRAM components soldered directly to PCB
  - Package-on-Package (PoP) devices
- 1, or 2 memory ranks comprised of x16 and/or x8 (“byte mode”) DRAM devices, including packages using x8 devices on one rank and x16 devices on another
  - In case of Mixed package, PHY supports only Byte-mode RL value (MR0 OP[1]=1) in both rank0/1
- PHY independent, firmware-based training using an embedded microcontroller
- Utilizes specialized hardware acceleration engines
- Automatic periodic retraining through DFI PHY Master Interface
  - PHY may initiate this training based on an internal timer
- Controller may also initiate periodic retraining directly via DFI frequency change protocol (without changing frequencies, if desired)
- LVSTL IO calibration and ODT calibration with:
- Pullup drive, pulldown drive, and pullup, pulldown ODT settings for DQ/DMI/parity bits
- Pullup drive, pulldown drive, and pullup, pulldown ODT settings for differential DQS/ WCK(LP5) pairs
- Pullup drive and pulldown drive for differential CK pairs
- Pullup drive and pulldown drive for CA/CS
  - Supports RZQ/N drive strength and termination down to a 40 ohm for LPDDR5 driver
  - Multiple drive strength settings for CMOS outputs
- Support of 4 PStates:
  - Can be trained upto 4 Power States (PStates).
  - PHY PStates contain fully-independent PHY control settings for each frequency, including all impedance, VREF values, and timing information
  - Fully-independent PHY control settings for each Pstate, including all impedances, VREF values, and timing information
  - Each PStates is associated with a fixed dfi\_freq\_ratio, trained voltage and temperature.
  - Each trained state is maintained across voltage and temperature variation using Periodic Phase Training (PPT)
- DFI interfaces are used during Periodic Phase Training.
  - Synopsys proprietary DFI Snoop Interface to detect and compensate voltage and temperature drift in LPDDR5 DRAM timing parameters (tWCK2DQI & tWCK2DQO)

- ❑ Frequency changes are initiated by the DFI interface without software involvement
- Three inactive idle states:
  - ❑ DFI\_LP Mode: most clocks and delay lines gated, including clock gating the majority of PUB logic. Includes several different power savings options, with differing wakeup times:
  - ❑ PHY Inactive (LP2): leakage only
  - ❑ PHY Retention (LP3/IO Retention): Core power removed, most I/Os powered down, SDRAMs held in self-refresh and Power-down
- Voltage and temperature compensated delay lines used for:
  - ❑ Centering each DQ/DMI/CA bit's eye around a clock/strobe
  - ❑ Read/write leveling
- Includes a low-jitter PLL for both PHY clock generation and SDRAM clock generation
- Only one PLL is required per DDR PHY, supporting 667 Mbps to 9600 Mbps operation
- Integrated PLL bypass to support 50 to 3200 Mbps
- Support for a SW controllable DQ bit and CA bit swizzling
- SW-defined mapping of CA bits within a channel
- SW-defined mapping of DQ bits within a byte
- SW-defined byte swap support for swapping 2 bytes within a 16-bit DRAM channel for PHY training purposes only
- It is up to the customer's system to choose the correct byte for DFI Command such as MRR when bytes are swapped on the board
- Support for 5-nm and below poly orientation rules
- Macrocells can be instantiated on orthogonal sides: Support for both East-West and North-South orientations
- Includes the PHY Utility Block (PUB)
- Soft IP Verilog design that includes PHY control features, such as read/write leveling and TUB for data training
- APB and JTAG interfaces for register access
- DFI interface for communicating with controller
- Test support:
  - ❑ Core-side and Pad-side loopback testing on both the address and data channels
    - Core-side loopback can be configured to bypass all I/O pad and prevent I/O toggle
    - ❑ Delay line BIST via built-in macro delay line oscillator
    - ❑ MUX-scan ATPG (stuck-at SCAN)
    - ❑ At-speed SCAN capture via built-in macro On-chip Clock Controller (OCC)
    - ❑ RX and TX asynchronous paths support a maximum bit rate of 600 Mb/s (300MHz) when controlled via bypass interface
    - ❑ PLL lock test
    - ❑ ZQ calibration test

- ❑ BurnIn test
- ❑ TDR interface to PHY registers (and SRAMs mapped to PHY register map) for easy test access
- ❑ Firmware-based 2D eye mapping diagnostic tool allows measuring 2D eye for every DQ bit of the bus at both DRAM and host receivers
- ❑ Direct override programming available for all VREF, ODT, drive strength, and timing delays to facilitate debug and characterization

## 1.5.2 LPDDR5/5x-Specific Features

Support for the following training features:

- Command Bus delay training for CA relative to CK, including CA bit deskew at user-selected CA-VREF
- Write Leveling to compensate for CK-DQS timing skew
- Write Training:
  - ❑ Per-bit DQS to DQ centering including compensation of DRAM tDQS2DQ delay
  - ❑ Per-rank VREFDQ training on DRAM DQ bits
  - ❑ Per DRAM-side DFE training
  - ❑ PUB will train DRAM-side WCK duty cycle correction (DCC).
- Read training:
  - ❑ Per bit DQS to DQ eye centering training using a combination of DRAM array and MPCs
  - ❑ Data bus VREFDQ training on each PHY DQ per bit
  - ❑ Host DFE will be trained as a part of 2-D read eye training
  - ❑ PHY Read DCA
- All JEDEC-defined DRAM addressing modes
- 1 or 2 memory ranks comprised of x16 and/or x8 (“byte mode”) DRAM devices, including packages using x8 devices on one rank and x16 devices on another
- Data bus inversion, data mask, DMI, link protection, and low power data copy supported in pass-through mode (controller to encode/decode)
- DVFS and DVFSQ supported, if enabled in the platform
  - ❑ Low standby and VDDQ signaling power down to 0.3 V operation. Max data rate for VDDQ = 0.3 V is system (channel) dependent (< 3200 Mbps).
- Standard DFI Interface
- LPDDR5 1:1:4 DfiClk:CK:WCK operation at boot time
- LPDDR5 1:1:2 DfiClk:CK:WCK operation at boot time.
- Supports fast FSP switch for LPDDR5 between CK: WCK 1:2 and 1:4 mode without the need to change DfiClk or lock the PLL again
- LPDDR5 CK:WCK ratio must always match the number of data phases in use on the DFI bus.
- DFI bus frequency must always match DRAM CK frequency in LPDDR5 mode.
- Supports fast FSP switch for LPDDR5 between CK: WCK 1:2 and 1:4 mode without the need to change DfiClk or lock the PLL again

- Automatic training of DQ VREF level setting separately for PHY per bit and for SDRAM per channel per rank
- LPDDR5 support for 2 tWCK Static, 2tWCK toggle read preamble only
- LPDDR5 support for 2.5 tWCK or 4.5 tWCK toggle read postamble only
- LPDDR5 supports only Toggle Mode for RDQS Postamble.
- Periodic retraining for DRAM write (tWCK2DQI) and read (tWCK2DQO) drift
  - MC initiated, using DFI Frequency Change Interface
  - MC initiated, using DFI Snoop (Synopsys proprietary) Interface
  - PHY initiated, using DFI PHY Master Interface
- By default, read data capture using differential RDQS strobe pair
- Single-Ended mode Support
  - When DRAM supports it, PHY can support single-ended mode on CK, WCK, and read DQS.
- Support for strobe-less read mode to capture read data at speed < = 1600 Mbps.

### 1.5.3 LPDDR4X-Specific Features

- Support for the following training features:
  - Command Bus delay training for CA relative to CK, including CA bit deskew at user-selected CA-VREF
  - Write Leveling to compensate for CK-DQS timing skew
  - Write Training:
    - Per-bit DQS to DQ centering including compensation of DRAM tDQS2DQ delay
    - Per-rank VREFDQ training on DRAM DQ bits
  - Read training:
    - Per bit DQS to DQ eye centering training using a combination of DRAM array and MPCs
    - Data bus VREFDQ training on each PHY DQ per bit
    - All JEDEC-defined DRAM addressing modes
    - Single-Ended mode Support
 

When DRAM supports it, PHY can support single-ended mode on CK, write DQS, and read DQS.

LPDDR4X DRAMs support single-ended mode only at 1600 Mbps or lower
- LPDDR4X support for all write preamble settings defined by JEDEC
- LPDDR4X support for all write postamble settings defined by JEDEC
- LPDDR4X support for toggling 2 tCK read preamble only
- LPDDR4X support for toggling 1.5 tCK read postamble only
- Data bus inversion and DMI supported in pass-through mode (controller to encode/decode)
- Periodic retraining for DRAM write (tDQS2DQ) and read (tDQSCK) drift

- DFI Interface:
  - DFI may select 1:2 or 1:4 mode at boot time
  - DFI is FSP-switchable between 1:2 and 1:4. This will enable a fast switch from LPDDR4X-N to LPDDR4X-N/2 without a PLL relock.
- DRAM device support:
  - Support all LPDDR4X compliant devices, including those following the addendum for byte-mode and single die-per-channel devices
    - DRAM device limitation of sharing single ZQ resistor across both channels prevent single die-per-channel LPDDR4X DRAM devices from being used if both DRAM channels are connected to a single set of CA/CS PHY output pins.
  - DRAM packages that mix x16 devices on one rank with byte-mode devices on another rank are supported

#### 1.5.4 I/O Features

- LPDDR5X and LPDDR4X operating modes.
- LPDDR5 DRAM DVFSQ support when external VDDQ adjustment is provided
- I/Os are integrated within the PHY Macrocells
- I/Os include receiver decision feedback equalization), supported for data rates > 4267 Mbps
- I/O includes transmitter pre-emphasis
- I/Os include transmitter time-domain DCA feature
- Programmable input on-die termination (ODT)
- Programmable output impedance with slew rate options
- Programmable DFE coefficients to support 1 to 2 tap DFE
- PVT-compensated ODT and output impedance
- Receiver I/O power-down control
- Embedded boundary scan support logic and bypass with secondary input & output ports
- Includes support logic for implementation JTAG SAMPLE instruction during mission-mode operation
  - Since receivers include samplers, non-mission-mode sampling uses internally generated clock per slice.
- PAD and internal loopback modes
- Supports flip chip packaging
- Retention IO added to enable full powerdown of VDD without power sequence issues during retention entry/exit
- Power on clear (POC) function renders library power supply sequence-agnostic
- SnapCap or MiMCap cells to support VDDQ-VSS decoupling capacitance

## 1.6 Compatible Standards

The following standards are compatible with the design.

**Table 1-1      Compatible Standards**

Standard	Compatible Version
DFI specification	V5.0
AMBA APB protocol	V2.0
JEDEC LPDDR4X SDRAM Specification	LPDDR4 JESD209-4D specification Addendum LPDDR4X JESD209-4-1A
JEDEC LPDDR5/5X SDRAM Specification	LPDDR5 JESD209-5B specification

## 1.7 Supported System Configurations

The list of all valid and supported configuration is shown in

### 1.7.1 PHY AC Configuration

LPDDR5X/5/4X PHY support following AC wrapper to support single channel as well dual-channel protocols.

**Table 1-2 Single Channel AC Configuration**

RTL Bump Name	LPDDR5X/5 Pins(2 ranks)	LPDDR4X Pins(2 ranks)	Hard Macro
BP_A[0]	CA0_A	CA0_A	ACX2_0
BP_A[1]	CA1_A	CA1_A	
BP_A[2]	CA2_A	CA2_A	ACX2_1
BP_A[3]	CA3_A	CA3_A	
BP_A[4]	CS0_A	CKE0_A	CSX2_0
BP_A[5]	CS1_A	CKE1_A	
BP_A[6]	CA4_A	CA4_A	ACX2_2
BP_A[7]	CA5_A	CA5_A	
BP_A[8]	CA6_A	CS0_A	ACX2_3
BP_A[9]	MTEST	CS1_A/MTEST	
BP_CK0_T	CK0_A_T	CK0_T	CKX2_0
BP_CK0_C	CK0_A_C	CK0_C	
BP.DTO0	MTEST	MTEST	ACX2_8
BP.DTO1	PllDigTst[1]	PllDigTst[1]	

Notes:

- Bump BP.DTO0/1 exists only when DWC\_LPDDR5XPHY.DTO\_ENABLED is defined.
  - MTEST is mapped to BP.DTO0 and PllDigTst[1] is mapped to BP.DTO1 bump.
  - MTEST is mapped to BP\_A[9] when DWC\_LPDDR5XPHY.DTO\_ENABLED is NOT defined.
- Any UNUSED IO should be in Low Power State, see “[Low Power Design](#)” on page [234](#).

**Table 1-3 Dual Channel AC Configuration**

<b>RTL Bump Name</b>	<b>LPDDR5X/5 Pins (2 ranks)</b>	<b>LPDDR4X Pins (2 ranks)</b>	<b>Hard Macro</b>	<b>HMAC IDX</b>
BP_A[0]	CA0_A	CA0_A	ACX2_0	0
BP_A[1]	CA1_A	CA1_A		
BP_A[2]	CA2_A	CA2_A	ACX2_1	1
BP_A[3]	CA3_A	CA3_A		
BP_A[4]	CS0_A	CKE0_A	CSX2_0	4
BP_A[5]	CS1_A	CKE1_A		
BP_A[6]	CA4_A	CA4_A	ACX2_2	2
BP_A[7]	CA5_A	CA5_A		
BP_A[8]	CA6_A	CS0_A	ACX2_3	3
BP_A[9]	MTEST <sup>2</sup>	CS1_A/MTEST		
BP_CK0_T	CK0_A_T	CK0_T	CKX2_0	5
BP_CK0_C	CK0_A_C	CK0_C		
BP_CK1_T	CK1_B_T	CK1_T	CKX2_1	11
BP_CK1_C	CK1_B_C	CK1_C		
BP_A[10]	CA0_B	CA0_B	ACX2_4	6
BP_A[11]	CA1_B	CA1_B		
BP_A[12]	CA2_B	CA2_B	ACX2_5	7
BP_A[13]	CA3_B	CA3_B		
BP_A[14]	CS0_B	CKE0_B	CSX2_1	10
BP_A[15]	CS1_B	CKE1_B		
BP_A[16]	CA4_B	CA4_B	ACX2_6	8
BP_A[17]	CA5_B	CA5_B		
BP_A[18]	CA6_B	CS0_B	ACX2_7	9
BP_A[19]	PllDigTst[1]	CS1_B/PllDigTst[1]		
BP.DTO0 <sup>1</sup>	MTEST	MTEST	ACX2_8	12
BP.DTO1	PllDigTst[1]	PllDigTst[1]		

## Notes:

1. Bump BP.DTO0/1 exists only when DWC\_LPDDR5XPHY.DTO\_ENABLED is defined.
  - BP.DTO is 1 bit wide. Other IO inside ACX2\_8 is unused and not connected to any BUMP.
  - MTEST is mapped to BP.DTO0 bump.
  - PllDigTst[1] is mapped to BP.DTO1 bump.
2. MTEST is mapped to BP.A[9] when DWC\_LPDDR5XPHY.DTO\_ENABLED is NOT defined.
3. PllDigTst[1] is mapped to BP.A[19] when DWC\_LPDDR5XPHY.DTO\_ENABLED is NOT defined
4. Any UNUSED IO should be in Low Power State, see “[Low Power Design](#)” on page [234](#).

Based on the bump assignments associated with specific hard macro instances, the location of certain hard macro cells need to be at specific places in the floorplan. For supported Floorplans in standard product, refer to Implementation Guide.

### 1.7.2 PHY DATA Configuration

**Table 1-4 DBYTE Configuration (DWC\_LPDDR5XPHY\_DBYTE\_DMI\_ENABLED is defined)**

RTL Bump Name	LPDDR5 Pins	LPDDR4X Pins	Hard Macro Bump	Hard Macro
BP_B*_D[0]	DQ0	DQ0	BP_DQ0	DX4
BP_B*_D[1]	DQ1	DQ1	BP_DQ1	
BP_B*_D[2]	DQ2	DQ2	BP_DQ2	
BP_B*_D[3]	DQ3	DQ3	BP_DQ3	
BP_B*_D[9]	DQS_T	DQS_T	BP_DQS_T	DX5
BP_B*_D[8]	DQS_C	DQS_C	BP_DQS_C	
BP_B*_D[11]	WCK_T	UNUSED	BP_DQS_T	
BP_B*_D[10]	WCK_C	UNUSED	BP_DQS_C	
BP_B*_D[12]	DMI	DM	BP_DQ4	
BP_B*_D[4]	DQ4	DQ4	BP_DQ0	
BP_B*_D[5]	DQ5	DQ5	BP_DQ1	
BP_B*_D[6]	DQ6	DQ6	BP_DQ2	
BP_B*_D[7]	DQ7	DQ7	BP_DQ3	

## 1.8 PUB Area

**Table 1-5** reflects approximate area estimates for the PUB. Actual cell counts vary based on clock frequency, technology, standard cell library, physical design constraints, synthesis/physical design tools, and methodology.

**Table 1-5** PUB Area

Configuration	Number of Registers (PUB)	Number of Combinational cells (PUB)	Number of Placeable components (PUB)	Number of equivalent 2-input NAND Gates	Estimated area (mm <sup>2</sup> ) in 5nm process PUB only	Estimated area (mm <sup>2</sup> ) in 5nm process phy_top (PUB+HM)	Estimated area (mm <sup>2</sup> ) Examples in 5nm at 0.5/0.4/0.2 utilization		
lp5x4xcs2dq18ch1	99354	434204	533558	1321880	0.042/utilization	0.487/utilization	0.084	0.105	0.21
lp5x4xcs2dq18ch2	148250	644287	792537	1958263	0.063/utilization	0.843/utilization	0.126	0.158	0.315

Synthesis results were generated using a sample configuration. Results may vary. Estimated area unit is mm<sup>2</sup>.

Utilization is the ratio of “pure synthesized gate area” / “final placement area”, see “[PUB Utilization](#)” on page 58, for an explanation

The examples assume 3 numbers for utilization, each represents a possible range

0.5 – Customers focused on minimizing area, doing a subsystem hardening can achieve a utilization between 0.4 and 0.55

0.4 – Customers focused on minimizing area, hardening the PHY separately can achieve a utilization between 0.3 and 0.45

0.2 – Customers with multiple construction restrictions (discussed below) may achieve a utilization between 0.05 and 0.3

## 1.9 PUB Utilization

The Synopsys DDR PHY is very configurable, allowing it to be manipulated to fit the needs of each customers SoC. PUB area will change dramatically as it is adapted into the application. Each of the considerations below will impact the utilization ratio of the PUB logic:

- Hardening shape and hierarchy
  - The DDR PHY typically has many bumps spread linearly along the beachfront of an SoC this will force the PUB to be long and thin. Buffering will be needed to distribute the signals across the PHY which will add to the gate count
  - Hardening the PHY and the memory controller in the same hierarchy will allow the logic to pack more tightly and improve the utilization
  - Some PHYs beachfront is limited by the bumps and requires the hard macros to be spaced out. This will grow the PUB area without changing the gate count, reducing the utilization
- Performance
  - High data rate PHYs may need additional register replication and pipe-lining to ease timing closure
  - Decoupling capacitors are needed to meet the VDD supply ripple specification. These will be added in the PUB using standard-cell decoupling capacitors and will impact utilization
  - High data rate PHYs may need additional buffering and larger output drive devices
- General
  - Clock tree buffers are not included in digital logic area estimates
  - Local decoupling capacitors are commonly added by modern placement tools to alleviate local IR drop violations
  - Areas of high power or reduced power grids will need the utilization reduced to avoid IR violations
  - Wide routing buses cause regions to be routing limited which reduces utilization
  - Projects where the full bump field must be within the bounding box of the PHY will artificially reduce utilization
  - Standard cell library choice. Performance, density targeted libraries will change utilization

## 1.10 Timing Closure

The PHY can accommodate a variety of compile options to allow the user to set the desired configuration and functionality. PHY also supports configurable pipeline options.

The DDR PHY also permits the user to implement various floor plan styles. This DDR PHY is designed for very high data rates and the maximum attainable performance may be affected by the user's selected floor plan and PUB compile options. Enabling both a wide interface and many optional features simultaneously may result in timing closure difficulties. A non-optimal floor plan, especially for a very wide interface, may also impact timing closure capability.

It is strongly recommended the user perform preliminary physical-based synthesis using EDA tools such as Synopsys' Design Compiler Graphical to estimate the timing closure capability of both the estimated floor plan and the selected DDR PHY configuration at the start of the project before deciding on the desired floor plan and compile options for the DDR PHY. Furthermore, in order to vet in advance possible timing closure issues, margining should be imposed during synthesis so as to emulate non-ideal circumstances (for example, clock skew, parasitics that vary from those described in wire models, path length-dependent derates, etc.) that will likely be encountered during downstream construction activities.

## 1.11 Limitation and Restrictions

Following protocols are not supported

- DDR3, DDR4 and DDR5 protocols are not supported
- LPDDR4 is not supported

## 1.12 Definition of Terms and Acronyms

**Table 1-6 Terms and Acronyms**

Term	Description
AC	Address/command
ACSM	Address Command State Machine. Used by Training Firmware
CSR	Control and Status Registers
DFI	DDR PHY Interface
DFICLK	DDR PHY Interface Clock Clock reference for all mission mode interface signals.
DTSM	Data Training State Machine
MC	Memory Controller – Interfaces with the PHY to provide operational commands, address, and data.
FSP	Frequency Set Point – Stored, trained values for specific operational frequencies
PMU	PHY micro-controller unit, same as uCtl
PPGC	PRBS Pattern Generator and Checker
PPT	Periodic Phase Training
PRBS	Pseudo Random Binary Sequence
PState	The PHY supports multiple power states, both Active and Standby. A power state (or PState) is a combination of memory clock frequency, driver strengths, termination, timing, and other CSR-enabled settings. The PHY supports 4 active power states. The PHY also supports 5 standby (inactive) power states, labeled DFI LP (3 states) Fast Standby (LP2), and IO Retention (LP3). Active power states are changed by utilizing the DFI Frequency Change protocol, via the "dfi_init_start", "dfi_frequency[]", and "dfi_init_complete" signals.
PUB	PHY Utility Block
TUB	Training Utility Block
uCtl	Micro-Controller
HM	Hard macro
HMAC	Hard macro Address/Command Block
HMDBYTE	Hard macro Data pipe Block
HMZCAL	Hard macro ZCAL Block
HMPAC	Hard macro PLL Analog Control Block

**Table 1-7 Hard Macro Mapping Table**

IP Block Name	Short Name	IP Type
dwc_lpddr5xphy_top	PHY	Soft IP
dwc_lpddr5xphyacx2_top_ew	ACX2	Hard macro
dwc_lpddr5xphyckx2_top_ew	CKx2	Hard macro
dwc_lpddr5xphycsx2_top_ew	CSX2	Hard macro
dwc_lpddr5xphycmosx2_top_ew	CMOS	Hard macro
dwc_lpddr5xphydx4_top_ew	DX4	Hard macro
dwc_lpddr5xphydx5_top_ew	DX5	Hard macro
dwc_lpddr5xphymaster_top_ew	PAC	Hard macro
dwc_lpddr5xphyzcal_top_ew	ZCAL	Hard macro
dwc_lpddr5xphy_decapvddq_x2_cell_ew	VDDQ DECAP CELL (Snapcap)	Utility Blocks
dwc_lpddr5xphy_decapvddq_hd_x2_cell_ew	VDDQ DECAP CELL (Snapcap)	
dwc_lpddr5xphy_decapvdd2h_x2_cell_ew	VDD2H DECAP CELL (Snapcap)	
dwc_lpddr5xphy_decapvdd_x2_cell_ew	VDD DECAP CELL (Snapcap)	
dwc_lpddr5xphy_decapvdd_hd_x2_cell_ew	VDD DECAP CELL (Snapcap)	
dwc_lpddr5xphy_decapvddq_x2_cell_ns	VDDQ DECAP CELL (Snapcap)	
dwc_lpddr5xphy_decapvddq_hd_x2_cell_ns	VDDQ DECAP CELL (Snapcap)	
dwc_lpddr5xphy_decapvddq_id_x2_cell_ns	VDDQ DECAP CELL (Snapcap)	
dwc_lpddr5xphy_decapvdd2h_x2_cell_ns	VDD2H DECAP CELL (Snapcap)	

IP Block Name	Short Name	IP Type
dwc_lpddr5xphy_decapvdd2h_id_x2_cell_ns	VDD2H DECAP CELL (Snapcap)	Utility Blocks
dwc_lpddr5xphy_decapvdd_x2_cell_ns	VDD DECAP CELL (Snapcap)	
dwc_lpddr5xphy_decapvdd_id_x2_cell_ns	VDD DECAP CELL (Snapcap)	
dwc_lpddr5xphy_decapvdd_hd_x2_cell_ns	VDD DECAP CELL (Snapcap)	
dwc_lpddr5xphy_vddqclamp_x2_ew	VDDQ CLAMP CELL	
dwc_lpddr5xphy_vdd2hclamp_x2_ew	VDD2H CLAMP CELL	
dwc_lpddr5xphy_pclk_rptx1	PCLK RPTX1	
dwc_lpddr5xphy_decapvddq_x2_ew	VDDQ DECAP X2 (Snapcap)	
dwc_lpddr5xphy_decapvddq_id_x2_ew	VDDQ DECAP X2 (Snapcap)	
dwc_lpddr5xphy_decapvddq_hd_x2_ew	VDDQ DECAP X2 (Snapcap)	
dwc_lpddr5xphy_decapvddq_x3_ew	VDDQ DECAP X3 (Snapcap)	
dwc_lpddr5xphy_decapvddq_id_x3_ew	VDDQ DECAP X3 (Snapcap)	
dwc_lpddr5xphy_decapvddq_hd_x3_ew	VDDQ DECAP X3 (Snapcap)	
dwc_lpddr5xphy_decapvdd_x2_ew	VDD DECAP X2 (Snapcap)	
dwc_lpddr5xphy_decapvdd_id_x2_ew	VDD DECAP X2 (Snapcap)	
dwc_lpddr5xphy_decapvdd_hd_x2_ew	VDD DECAP X2 (Snapcap)	
dwc_lpddr5xphy_decapvdd_x3_ew	VDD DECAP X3 (Snapcap)	
dwc_lpddr5xphy_decapvdd_id_x3_ew	VDD DECAP X3 (Snapcap)	
dwc_lpddr5xphy_decapvdd_hd_x3_ew	VDD DECAP X3 (Snapcap)	
dwc_lpddr5xphy_decapvdd2h_x2_ew	VDD2H DECAP X2 (Snapcap)	Utility Blocks
dwc_lpddr5xphy_decapvdd2h_id_x2_ew	VDD2H DECAP X2 (Snapcap)	
dwc_lpddr5xphy_decapvdd2h_x3_ew	VDD2H DECAP X3 (Snapcap)	
dwc_lpddr5xphy_decapvdd2h_id_x3_ew	VDD2H DECAP X3 (Snapcap)	
dwc_lpddr5xphy_vdd2hclamp_ew	VDD2H CLAMP	
dwc_lpddr5xphy_vdd2hclamp_x6_ew	VDD2H CLAMP X6	
dwc_lpddr5xphy_pclk_routing_ac_ew	Pclk Routing AC	
dwc_lpddr5xphy_pclk_routing_dx_ew	Pclk Routing DX	
dwc_lpddr5xphy_pclk_routing_decapvddq_ew	Pclk Routing Decap	
dwc_lpddr5xphy_pclk_routing_decapvdd_ew	Pclk Routing Decap	

**Table 1-8 Custom Circuit Macro Mapping Table**

IP Block Name	Short Name	IP Type
dwc_lpddr5xphy_txfe	TXFE	Custom Circuit Macro
dwc_lpddr5xphy_dcd	DCD	
dwc_lpddr5xphy_txbe	TXBE	
dwc_lpddr5xphy_txrxdq	TXRXDQ	
dwc_lpddr5xphy_txrxdqs	TXRXDQS	
dwc_lpddr5xphy_txrxac	TXRXAC	
dwc_lpddr5xphy_txrxcs	TXRXCS	
dwc_lpddr5xphy_txrxcmos	TXRXCMOS	
dwc_lpddr5xphy_rxreplica	RX Replica	
dwc_lpddr5xphy_rxdqs	RXDQS	
dwc_lpddr5xphy_rxac	RXAC	
dwc_lpddr5xphy_txfecs	TXFECS	
dwc_lpddr5xphy_txbecs	TXBECOS	
dwc_lpddr5xphy_rxcs	RXCS	
dwc_lpddr5xphy_por	POR	
dwc_lpddr5xphy_pclk_master	PCLK PAC	
dwc_lpddr5xphy_pclk_rptx1	PCLK RPTX1	
dwc_lpddr5xphy_pclk_rxda	PCLK RXDCA	
dwc_lpddr5xphy_lstx_(acx2 csx2 dx4 d x5 zal)	LSTX	

# 2

## Functional Overview

---

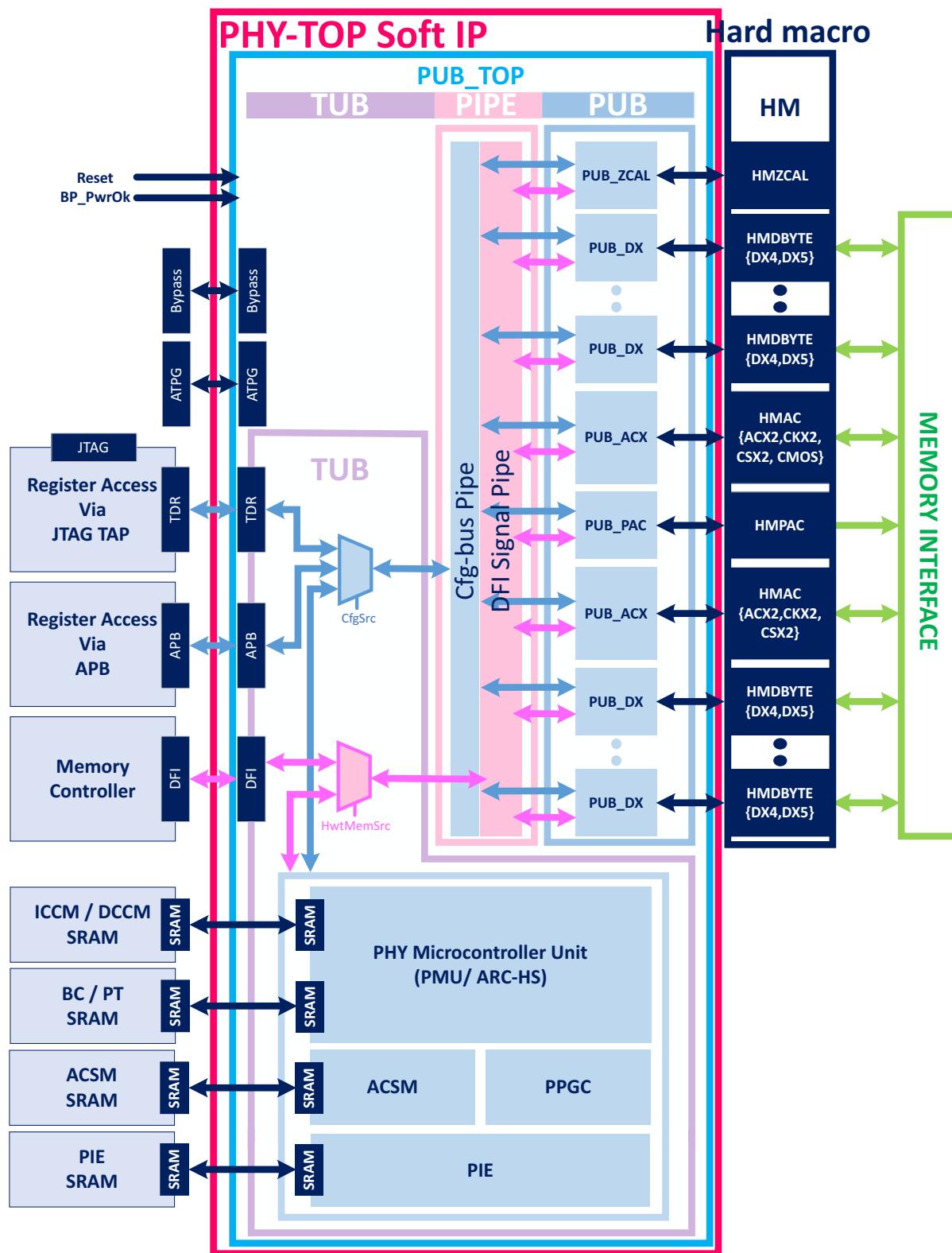
The following sections are included in this chapter:

- “[PUB Functional Overview](#)” on page [66](#)
- “[DFI Connections and System Memory Configurations](#)” on page [69](#)

## 2.1 PUB Functional Overview

The DWC PHY Utility Block (PUB) is a soft IP block that is used with the Synopsys PHY IP. It provides control features to ease the customer implementation of digitally controlled DDR PHY features such as read DQS training, data eye training, output impedance calibration. It also provides a DFI interface to the PHY. The PUB performs, in sequence, various tasks required by the PHY before it can commence normal DDR operations.

Figure 2-1 Functional Overview



SDRAM memory read/write access through the PHY is primarily through a DFI interface on the PUB. However, internal PUB logic can also access the SDRAM memory during training and diagnostic test procedures.



**Note** The memory controller used with the PHY must be DFI compatible.

Access to the PUB internal control features and registers is through a dedicated configuration port, the APB. A TDR interface is included as an additional second configuration port to co-exist with the APB configuration ports.

The PUB is driven off three clocks, the DFICLK, the APBCLK for APB interface, and the TDRCLK for TDR interface. The APBCLK drives the APB interface logic and is asynchronous to TDRCLK and DFICLK; The TDRCLK similarly drives JTAG interface logic and is asynchronous to APBCLK and DFICLK. All internal configuration registers, training and initialization logic, and PUB address/data paths are driven from the DFICLK.

The following is an overview of the different design blocks and functionality of the PUB:

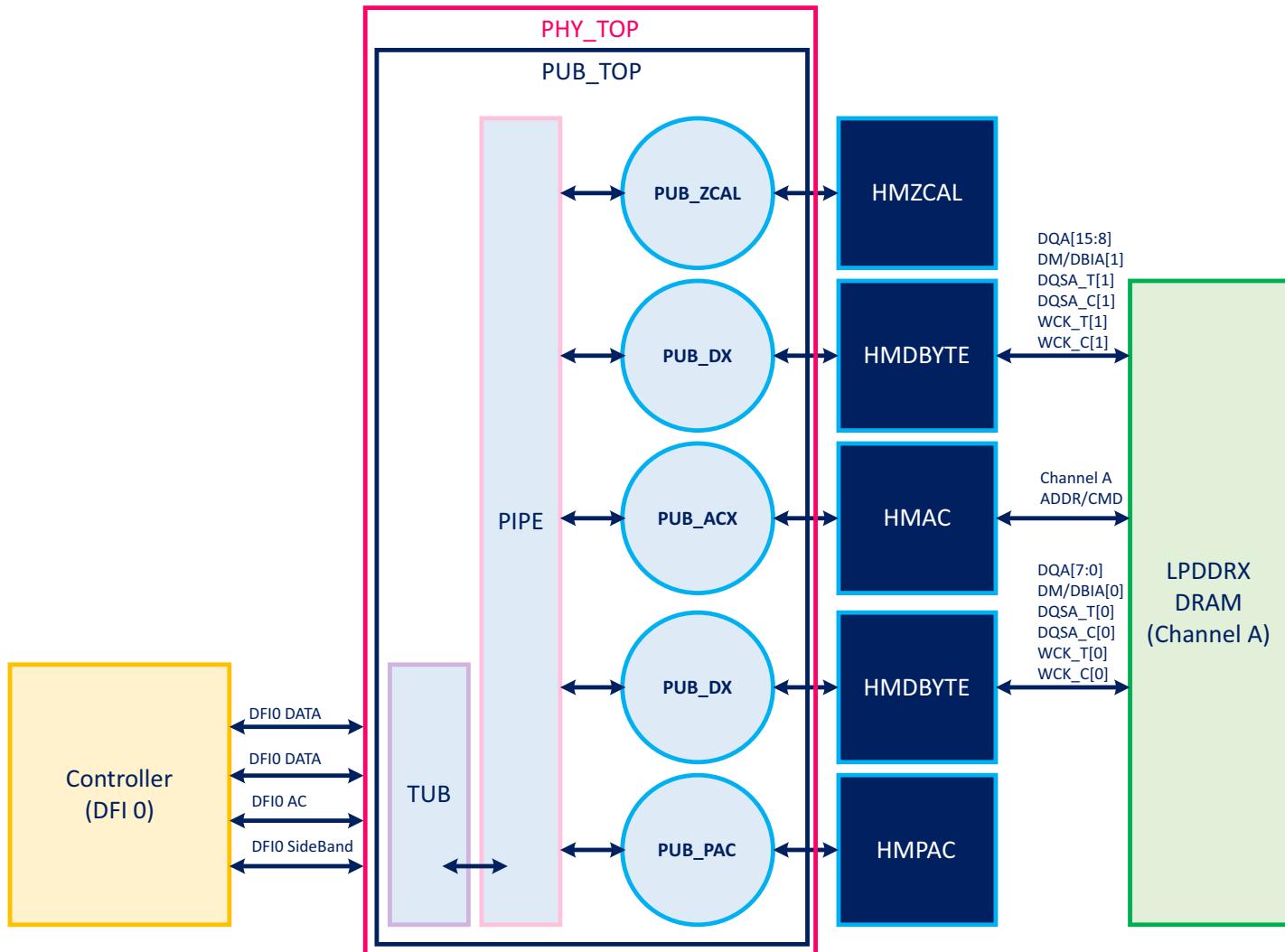
- Configuration Registers
  - Includes all registers required to configure, control, train and self-test the PHY. The configuration block also contains registers required to initialize the DRAM.
- PHY Microcontroller Unit (PMU)
  - Initialization
  - Used to initialize the PHY, including locking of the PLL, delay line calibration and calibration of I/O impedances.
- Data Training
  - Used to train the PHY for optimum timing margins. This includes finding the best positioning of the DQS gating window during reads, write leveling, data bit deskew, and optimizing the read and write data eye.
- DFI Interface Block
  - Implements a DFI-compliant interface between the external controller and the PHY.

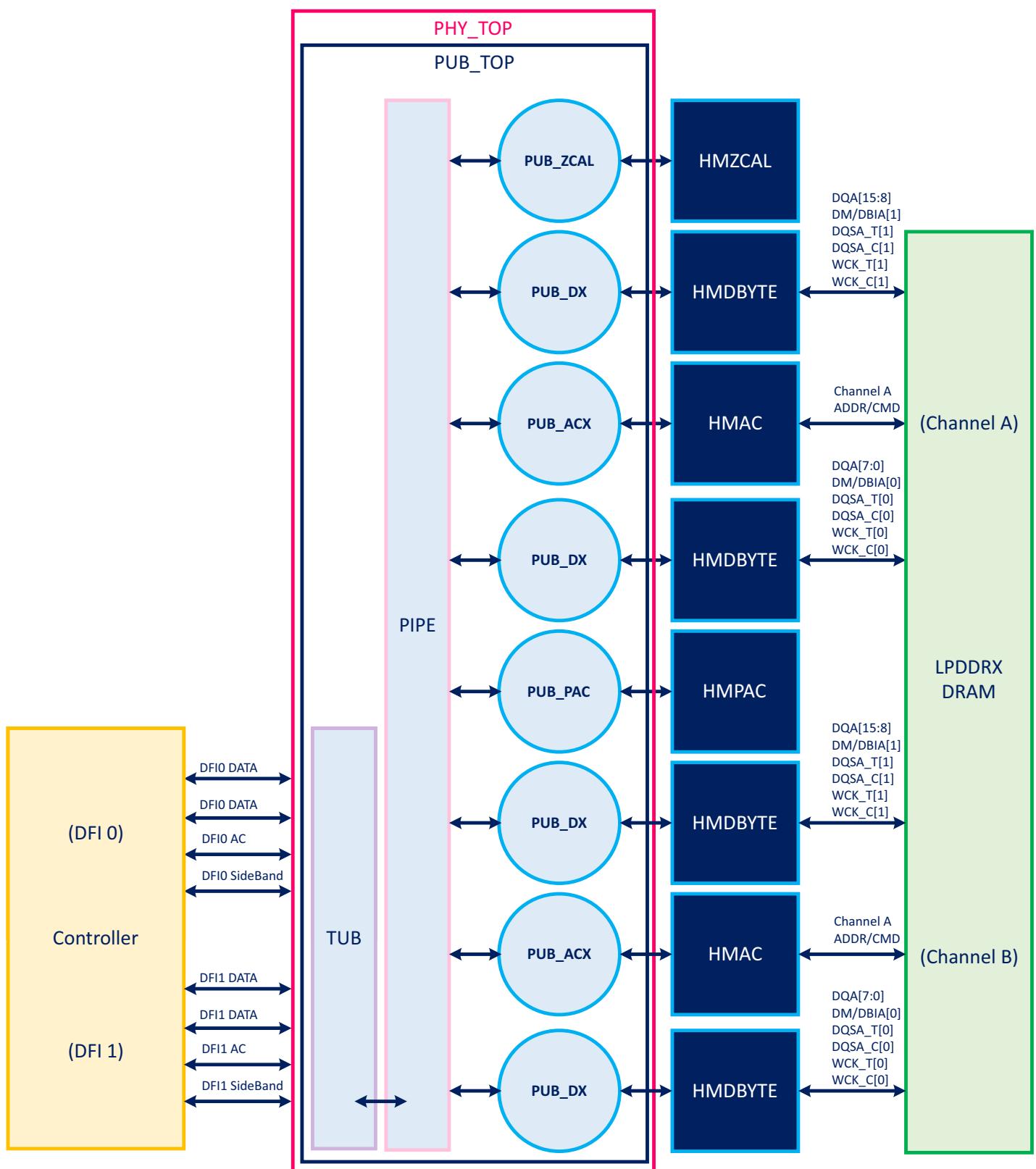
## 2.2 DFI Connections and System Memory Configurations

The PHY supports multiple configurations of system memory. The memory type, width, and number of channels used in the LP configuration determines the controller connections to and usage of the DFI bus. The following figures show examples of several memory configurations and their connections.

Below are supported configuration in LPDDR5X/5/4X PHY.

**Figure 2-2 Single Channel LPDDR DFI Connection**



**Figure 2-3** Dual Channel LPDDR DFI Connection

The 2 channels shown here could be operated independently, or with both controllers running in lock-step, or with a single controller operating both PHY channels in lock-step.



# 3

## Signal Descriptions

This chapter details all I/O signals in the IP. Inputs are on the left of the signal diagrams; outputs are on the right. In addition to describing the function of each signal, the signal descriptions in this chapter might include the following information:

- **Active State:** Indicates whether the signal is active high or active low. When a signal is not intended to be used in a particular application, then this signal needs to be tied or driven to the inactive state (opposite of the active state).
- **Registered:** Indicates whether or not the signal is registered directly inside the IP boundary without intervening logic (excluding simple buffers). A value of No does not imply that the signal is not synchronous, only that there is some combinatorial logic between the signal's origin or destination register and the boundary of the IP. A value of N/A indicates that this information is not provided for this IP title.
- **Synchronous to:** Indicates which clocks in the IP sample this input (drive for an output). This clock might not be the same as the clock that your application logic should use to clock (sample/drive) this pin. For more details, consult the clock section in the databook.

The I/O signals are grouped as follows:

- “DFI Signals” on page 75
- “Misc Signals” on page 83
- “Reset Signals” on page 84
- “PLL Clock Signals” on page 85
- “Test Signals” on page 86
- “TDR Signals” on page 87
- “Scan Signals” on page 89
- “AHB-Lite Signals” on page 91
- “APB Signals” on page 93
- “SRAM Signals” on page 96
- “Bypass mode Signals” on page 103
- “Bump Signals” on page 109
- “Power and Ground Signals” on page 111

## 3.1 DFI Signals

```

dfi0_wrdata_mask_Pz (for z = 0; z <= 3) -
dfi0_wck_write_Pz (for z = 0; z <= 3) -
dfi0_wck_en_Pz (for z = 0; z <= 3) -
dfi0_wck_cs_Pz (for z = 0; z <= 3) -
dfi0_wck_toggle_Pz (for z = 0; z <= 3) -
dfi0_wrdata_ecc_Pz (for z = 0; z <= 3) -
dfi0_address_Pz (for z = 0; z <= 3) -
dfi0_wrdata_Pz (for z = 0; z <= 3) -
dfi0_wrdata_cs_Pz (for z = 0; z <= 3) -
dfi0_wrdata_en_Pz (for z = 0; z <= 3) -
dfi0_rddata_cs_Pz (for z = 0; z <= 3) -
dfi0_rddata_en_Pz (for z = 0; z <= 3) -
dfi0_ctrlupd_req -
dfi0_ctrlupd_type -
dfi0_phyupd_ack -
dfi0_dram_clk_disable_Pz (for z = 0; z <= 3) -
dfi0_freq_fsp -
dfi0_freq_ratio -
dfi0_frequency -
dfi0_init_start -
dfi0_phymstr_ack -
dfi0_cke_Pz (for z = 0; z <= 3) -
dfi0_cs_Pz (for z = 0; z <= 3) -
dfi0_lp_ctrl_req -
dfi0_lp_ctrl_wakeup -
dfi0_lp_data_req -
dfi0_lp_data_wakeup -
dfi0_lp_reset_n -
dfi1_wck_write_Pz (for z = 0; z <= 3) -
dfi1_wck_en_Pz (for z = 0; z <= 3) -
dfi1_wck_cs_Pz (for z = 0; z <= 3) -
dfi1_wck_toggle_Pz (for z = 0; z <= 3) -
dfi1_wrdata_ecc_Pz (for z = 0; z <= 3) -
dfi1_address_Pz (for z = 0; z <= 3) -
dfi1_wrdata_mask_Pz (for z = 0; z <= 3) -
dfi1_wrdata_Pz (for z = 0; z <= 3) -
dfi1_wrdata_cs_Pz (for z = 0; z <= 3) -
dfi1_wrdata_en_Pz (for z = 0; z <= 3) -
dfi1_rddata_cs_Pz (for z = 0; z <= 3) -
dfi1_rddata_en_Pz (for z = 0; z <= 3) -
dfi1_ctrlupd_req -
dfi1_ctrlupd_type -
dfi1_phyupd_ack -
dfi1_dram_clk_disable_Pz (for z = 0; z <= 3) -
dfi1_freq_fsp -
dfi1_freq_ratio -
dfi1_frequency -
dfi1_init_start -
dfi1_phymstr_ack -
dfi1_cke_Pz (for z = 0; z <= 3) -
dfi1_cs_Pz (for z = 0; z <= 3) -
dfi1_lp_ctrl_req -
dfi1_lp_ctrl_wakeup -
dfi1_lp_data_req -
dfi1_lp_data_wakeup -
DfiClk -
- dfi0_rddata_db1_Wz (for z = 0; z <= 3)
- dfi0_rddata_Wz (for z = 0; z <= 3)
- dfi0_rddata_valid_Wz (for z = 0; z <= 3)
- dfi0_ctrlupd_ack
- dfi0_phyupd_req
- dfi0_phyupd_type
- dfi0_init_complete
- dfi0_phymstr_cs_state
- dfi0_phymstr_req
- dfi0_phymstr_state_sel
- dfi0_phymstr_type
- dfi0_lp_ctrl_ack
- dfi0_lp_data_ack
- dfi0_error
- dfi0_error_info
- dfi1_rddata_db1_Wz (for z = 0; z <= 3)
- dfi1_rddata_Wz (for z = 0; z <= 3)
- dfi1_rddata_valid_Wz (for z = 0; z <= 3)
- dfi1_ctrlupd_ack
- dfi1_phyupd_req
- dfi1_phyupd_type
- dfi1_init_complete
- dfi1_phymstr_cs_state
- dfi1_phymstr_req
- dfi1_phymstr_state_sel
- dfi1_phymstr_type
- dfi1_lp_ctrl_ack
- dfi1_lp_data_ack
- dfi1_error
- dfi1_error_info

```

**Table 3-1 DFI Signals**

Port Name	I/O	Description
dfi0_wrdata_mask_Pz[3:0] (for z = 0; z <= 3)	I	dfi0_wrdata_mask_P0: DFI write data byte mask <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_rddata_dbi_Wz[3:0] (for z = 0; z <= 3)	O	dfi0_rddata_dbi_W0: DFI read data DBI <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_wck_write_Pz[1:0] (for z = 0; z <= 3)	I	Reserved. Should be tied to 0. <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_wck_en_Pz[1:0] (for z = 0; z <= 3)	I	dfi0_wck_en_P0: DFI enable/disable control for WCK <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_wck_cs_Pz[3:0] (for z = 0; z <= 3)	I	dfi0_wck_cs_P0: DFI active rank control for WCK <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_wck_toggle_Pz[3:0] (for z = 0; z <= 3)	I	dfi0_wck_toggle_P0: DFI state control for WCK <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_wrdata_ecc_Pz[3:0] (for z = 0; z <= 3)	I	dfi0_wrdata_ecc_P0: Optional link ECC for LPDDR5 <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_address_Pz[13:0] (for z = 0; z <= 3)	I	DFI address bus <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_wrdata_Pz[31:0] (for z = 0; z <= 3)	I	dfi0_wrdata_P0: DFI write data <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_wrdata_cs_Pz[3:0] (for z = 0; z <= 3)	I	dfi0_wrdata_cs_P0: DFI write data chip select <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_wrdata_en_Pz[1:0] (for z = 0; z <= 3)	I	dfi0_wrdata_en_P0: DFI write data and data mask enable <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_rddata_Wz[31:0] (for z = 0; z <= 3)	O	dfi0_rddata_W0: DFI read data <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk

Port Name	I/O	Description
dfi0_rddata_cs_Pz[3:0] (for z = 0; z <= 3)	I	dfi0_rddata_cs_P0: DFI read data chip select <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_rddata_en_Pz[1:0] (for z = 0; z <= 3)	I	dfi0_rddata_en_P0: DFI read data enable <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_rddata_valid_Wz[1:0] (for z = 0; z <= 3)	O	dfi0_rddata_valid_W0: DFI read data valid indicator <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_ctrlupd_ack	O	dfi0_ctrlupd_ack: DFI memory controller initiated update acknowledge <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_ctrlupd_req	I	dfi0_ctrlupd_req: DFI memory controller initiated update request <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_ctrlupd_type[1:0]	I	DFI0 memory controller initiated update request <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_phyupd_ack	I	dfi0_phyupd_ack: DFI PHY initiated update acknowledge <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_phyupd_req	O	dfi0_phyupd_req-DFI PHY initiated update request <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_phyupd_type[1:0]	O	dfi0_phyupd_type: DFI PHY initiated update select <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_dram_clk_disable_Pz (for z = 0; z <= 3)	I	DFI DRAM clock disable for BP_DFI0_CK_T/C <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_freq_fsp[1:0]	I	dfi0_freq_fsp: DFI frequency <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_freq_ratio[1:0]	I	dfi0_freq_ratio: DFI frequency ratio indicator <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk

Port Name	I/O	Description
dfi0_frequency[4:0]	I	dfi0_frequency: DFI frequency <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_init_complete	O	dfi0_init_complete: DFI PHY initialization complete <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_init_start	I	dfi0_init_start-DFI setup stabilization or frequency change <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_phymstr_ack	I	dfi0_phymstr_ack: DFI PHY Master Acknowledge <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_phymstr_cs_state[(DWC_LPDDR5XPHY_DFI0_PHY_DFI0_PHYMSTR_CS_STATE_WIDTH-1):0]	O	dfi0_phymstr_cs_state: DFI PHY Master CS State <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_phymstr_req	O	dfi0_phymstr_req: DFI PHY Master Request <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_phymstr_state_sel	O	dfi0_phymstr_state_sel: DFI PHY Master State Select <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_phymstr_type[1:0]	O	dfi0_phymstr_type: DFI PHY Master Type <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_cke_Pz[(DWC_LPDDR5XPHY_DFI0_CKE_WIDTH-1):0] (for z = 0; z <= 3)	I	DFI clock enable. These pins are present only used when LPDDR4X DRAM is used <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_cs_Pz[(DWC_LPDDR5XPHY_DFI0_CS_WIDTH-1):0] (for z = 0; z <= 3)	I	DFI chip select <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_lp_ctrl_ack	O	dfi0_lp_ctrl_ack: DFI low power port control acknowledge <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_lp_ctrl_req	I	dfi0_lp_ctrl_req: DFI low power port control request <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk

Port Name	I/O	Description
dfi0_lp_ctrl_wakeup[4:0]	I	dfi0_lp_ctrl_wakeup: DFI wakeup from low power <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_lp_data_ack	O	dfi0_lp_data_ack: DFI low power port data acknowledge <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_lp_data_req	I	dfi0_lp_data_req: DFI low power port data request <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_lp_data_wakeup[4:0]	I	dfi0_lp_data_wakeup: DFI wakeup from low power <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_error	O	dfi0_error: DFI error <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi0_error_info[3:0]	O	dfi0_error_info: DFI error Info <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi_reset_n	I	DFI reset bus <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_wck_write_Pz[1:0] (for z = 0; z <= 3)	I	Reserved. Should be tied to 0. <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_wck_en_Pz[1:0] (for z = 0; z <= 3)	I	DFI enable/disable control for WCK <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_wck_cs_Pz[3:0] (for z = 0; z <= 3)	I	dfi1_wck_cs_P0: DFI active rank control for WCK <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_wck_toggle_Pz[3:0] (for z = 0; z <= 3)	I	dfi1_wck_toggle_P0: DFI state control for WCK <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_wrdata_ecc_Pz[3:0] (for z = 0; z <= 3)	I	dfi1_wrdata_ecc_P0: Optional link ECC for LPDDR5 <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk

Port Name	I/O	Description
dfi1_address_Pz[13:0] (for z = 0; z <= 3)	I	DFI address bus <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_wrdata_mask_Pz[3:0] (for z = 0; z <= 3)	I	DFI write DMI or DBI <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_rddata_dbz_Wz[3:0] (for z = 0; z <= 3)	O	dfi1_rddata_dbz_W0: DFI read data DBI <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_wrdata_Pz[31:0] (for z = 0; z <= 3)	I	dfi1_wrdata_P0: DFI write data <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_wrdata_cs_Pz[3:0] (for z = 0; z <= 3)	I	dfi1_wrdata_cs_P0: DFI write data chip select <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_wrdata_en_Pz[1:0] (for z = 0; z <= 3)	I	dfi1_wrdata_en_P0: DFI write data and data mask enable <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_rddata_Wz[31:0] (for z = 0; z <= 3)	O	dfi1_rddata_W0: DFI read data <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_rddata_cs_Pz[3:0] (for z = 0; z <= 3)	I	dfi1_rddata_cs_P0: DFI read data chip select <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_rddata_en_Pz[1:0] (for z = 0; z <= 3)	I	dfi1_rddata_en_P0: DFI read data enable <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_rddata_valid_Wz[1:0] (for z = 0; z <= 3)	O	dfi1_rddata_valid_W0: DFI read data valid indicator <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_ctrlupd_ack	O	dfi1_ctrlupd_ack: DFI memory controller initiated update acknowledge <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_ctrlupd_req	I	dfi1_ctrlupd_req: DFI memory controller initiated update request <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk

Port Name	I/O	Description
dfi1_ctrlupd_type[1:0]	I	DFI1 memory controller initiated update request <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_phyupd_ack	I	dfi1_phyupd_ack: DFI PHY initiated update acknowledge <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_phyupd_req	O	dfi1_phyupd_req-DFI PHY initiated update request <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_phyupd_type[1:0]	O	dfi1_phyupd_type: DFI PHY initiated update select <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_dram_clk_disable_Pz (for z = 0; z <= 3)	I	DFI DRAM clock disable for BP_DFI0_CK_T/C <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_freq_fsp[1:0]	I	dfi1_freq_fsp: DFI frequency <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_freq_ratio[1:0]	I	dfi1_freq_ratio: DFI frequency ratio indicator <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_frequency[4:0]	I	dfi1_frequency: DFI frequency <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_init_complete	O	dfi1_init_complete: DFI PHY initialization complete <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_init_start	I	dfi1_init_start-DFI setup stabilization or frequency change <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_phymstr_ack	I	dfi1_phymstr_ack: DFI PHY Master Acknowledge <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_phymstr_cs_state[1:0]	O	dfi1_phymstr_cs_state: DFI PHY Master CS State <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk

Port Name	I/O	Description
dfi1_phymstr_req	O	dfi1_phymstr_req: DFI PHY Master Request <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_phymstr_state_sel	O	dfi1_phymstr_state_sel: DFI PHY Master State Select <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_phymstr_type[1:0]	O	dfi1_phymstr_type: DFI PHY Master Type <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_cke_Pz[1:0] (for z = 0; z <= 3)	I	DFI clock enable. These pins are present only used when LPDDR4X DRAM is used <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_cs_Pz[1:0] (for z = 0; z <= 3)	I	DFI chip select <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_lp_ctrl_ack	O	dfi1_lp_ctrl_ack: DFI low power port control acknowledge <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_lp_ctrl_req	I	dfi1_lp_ctrl_req: DFI low power port control request <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_lp_ctrl_wakeup[4:0]	I	dfi1_lp_ctrl_wakeup: DFI wakeup from low power <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_lp_data_ack	O	dfi1_lp_data_ack: DFI low power port data acknowledge <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_lp_data_req	I	dfi1_lp_data_req: DFI low power port data request <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_lp_data_wakeup[4:0]	I	dfi1_lp_data_wakeup: DFI wakeup from low power <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dfi1_error	O	dfi1_error: DFI error <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk

Port Name	I/O	Description
dfi1_error_info[3:0]	O	dfi1_error_info: DFI error Info <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
DfiClk	I	Internal DDR PHY Interface Clock. Depending on the value of the data-rate of the DRAM bus <b>Exists:</b> Always <b>Synchronous To:</b> None

## 3.2 Misc Signals

dwc\_lpddr5xphy0\_snoop\_en\_Pz (for z = 0; z <= 3) -  
 dwc\_lpddr5xphy1\_snoop\_en\_Pz (for z = 0; z <= 3) -  
 dwc\_lpddr5xphy0\_snoop\_osc\_running -  
 dwc\_lpddr5xphy1\_snoop\_osc\_running -  
- dwc\_lpddr5xphy\_pll\_lock  
- PhyInt\_n  
- PhyInt\_fault  
- dwc\_lpddr5xphy\_pmu\_busy

**Table 3-2** Misc Signals

Port Name	I/O	Description
dwc_lpddr5xphy0_snoop_en_Pz[7:0] (for z = 0; z <= 3)	I	Channel-0 P0 DFI snoop (for ppt) <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dwc_lpddr5xphy1_snoop_en_Pz[7:0] (for z = 0; z <= 3)	I	Channel-1 P0 DFI snoop (for ppt) <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dwc_lpddr5xphy0_snoop_osc_running	I	indicates an osc running <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dwc_lpddr5xphy1_snoop_osc_running	I	indicates an osc running <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dwc_lpddr5xphy_pll_lock	O	PLL Lock observation <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
PhyInt_n[15:0]	O	Interrupt Output <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
PhyInt_fault[5:0]	O	Anti-Valent Interrupt Output <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dwc_lpddr5xphy_pmu_busy	O	RFU <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk

### 3.3 Reset Signals



Table 3-3 Reset Signals

Port Name	I/O	Description
Reset	I	PHY warm reset <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
Reset_async	I	Async Reset for HardIP <b>Exists:</b> Always <b>Synchronous To:</b> None

## 3.4 PLL Clock Signals



**Table 3-4 PLL Clock Signals**

Port Name	I/O	Description
PllRefClk	I	PLL Reference Clock <b>Exists:</b> Always <b>Synchronous To:</b> None
PllBypClk	I	PLL bypass Clock <b>Exists:</b> Always <b>Synchronous To:</b> None

## 3.5 Test Signals



**Table 3-5 Test Signals**

Port Name	I/O	Description
BurnIn	I	Active High ATPG burn in control input <b>Exists:</b> Always <b>Synchronous To:</b> None
dwc_lpddr5xphy(dto[1:0]	O	Digital Test Observability pin <b>Exists:</b> Always <b>Synchronous To:</b> None
Iddq_mode	I	Active High IDDQ control input <b>Exists:</b> Always <b>Synchronous To:</b> None

## 3.6 TDR Signals

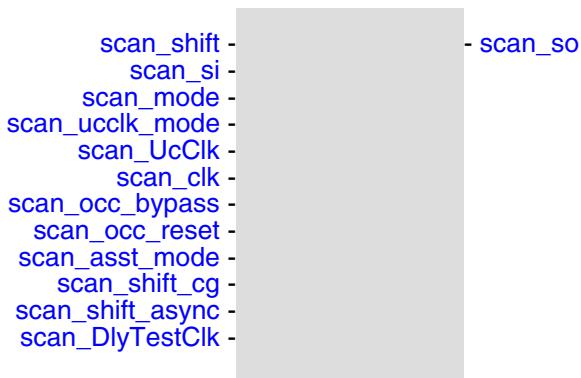


**Table 3-6 TDR Signals**

Port Name	I/O	Description
WSI	I	TDR Serial Data In <b>Exists:</b> Always <b>Synchronous To:</b> TDRCLOCK
TDRCLOCK	I	TDR clock <b>Exists:</b> Always <b>Synchronous To:</b> None
WRSTN	I	Low true reset to TDR <b>Exists:</b> Always <b>Synchronous To:</b> TDRCLOCK
DdrPhyCsrCmdTdrShiftEn	I	TDR register control signals used to have TDR set up CfgCmd for reads and writes of CSRs <b>Exists:</b> Always <b>Synchronous To:</b> TDRCLOCK
DdrPhyCsrCmdTdrCaptureEn	I	TDR capture enable for TDR CfgCmd for reads and writes of CSRs <b>Exists:</b> Always <b>Synchronous To:</b> TDRCLOCK
DdrPhyCsrCmdTdrUpdateEn	I	TDR update enable for TDR CfgCmd for reads and writes of CSRs <b>Exists:</b> Always <b>Synchronous To:</b> TDRCLOCK
DdrPhyCsrCmdTdr_Tdo	O	TDR output data for TDR CfgCmd for reads and writes of CSRs <b>Exists:</b> Always <b>Synchronous To:</b> TDRCLOCK
DdrPhyCsrRdDataTdrShiftEn	I	TDR register control signals used to have TDR read data back from Cfg bus to access CSRs <b>Exists:</b> Always <b>Synchronous To:</b> TDRCLOCK

Port Name	I/O	Description
DdrPhyCsrRdDataTdrCaptureEn	I	TDR capture enable for TDR Cfg for reads and writes of CSRs <b>Exists:</b> Always <b>Synchronous To:</b> TDRCLK
DdrPhyCsrRdDataTdrUpdateEn	I	TDR update enable for TDR Cfg for reads and writes of CSRs <b>Exists:</b> Always <b>Synchronous To:</b> TDRCLK
DdrPhyCsrRdDataTdr_Tdo	O	TDR output data for TDR Cfg for reads and writes of CSRs <b>Exists:</b> Always <b>Synchronous To:</b> TDRCLK

## 3.7 Scan Signals



**Table 3-7 Scan Signals**

Port Name	I/O	Description
scan_shift[5:0]	I	ATPG Scan Shift Enable <b>Exists:</b> Always <b>Synchronous To:</b> scan_clk
scan_si[(DWC_LPDDR5XPHY_NUM_TO_P_SCAN_CHAINS-1):0]	I	ATPG Scan Chain Input <b>Exists:</b> Always <b>Synchronous To:</b> scan_clk
scan_so[(DWC_LPDDR5XPHY_NUM_TO_P_SCAN_CHAINS-1):0]	O	ATPG Scan Chain Output <b>Exists:</b> Always <b>Synchronous To:</b> scan_clk
scan_mode	I	ATPG Scan Mode Enable <b>Exists:</b> Always <b>Synchronous To:</b> scan_clk
scan_ucclk_mode	I	ATPG Scan Mode Enable <b>Exists:</b> Always <b>Synchronous To:</b> scan_clk
scan_UcClk	I	atpg clock for UcClk clk <b>Exists:</b> Always <b>Synchronous To:</b> None
scan_clk	I	ATPG Scan slow clock <b>Exists:</b> Always <b>Synchronous To:</b> None
scan_occ_bypass	I	ATPG Scan OCC bypass enable <b>Exists:</b> Always <b>Synchronous To:</b> scan_clk

Port Name	I/O	Description
scan_occ_reset	I	OCC logic reset <b>Exists:</b> Always <b>Synchronous To:</b> scan_clk
scan_asst_mode	I	ATPG at-speed Scan Mode enabled <b>Exists:</b> Always <b>Synchronous To:</b> scan_clk
scan_shift_cg	I	ATPG Scan Shift Enable for clock-gaters <b>Exists:</b> Always <b>Synchronous To:</b> scan_clk
scan_shift_async	I	Scan shift enable for async bypass-mux <b>Exists:</b> Always <b>Synchronous To:</b> scan_clk
scan_DlyTestClk	I	scan_DlyTestClk: ATPG Scan slow clock <b>Exists:</b> Always <b>Synchronous To:</b> None

## 3.8 AHB-Lite Signals

hdata\_ahb -  
 hresp\_ahb -  
 hreadyout\_ahb -  
- haddr\_ahb  
- hburst\_ahb  
- hmastlock\_ahb  
- hprot\_ahb  
- hsize\_ahb  
- htrans\_ahb  
- hwdata\_ahb  
- hwrite\_ahb  
- hclk\_ahb  
- hresetn\_ahb

**Table 3-8 AHB-Lite Signals**

Port Name	I/O	Description
haddr_ahb[13:0]	O	haddr_ahb: RFU <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
hburst_ahb[2:0]	O	hburst_ahb: RFU <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
hmastlock_ahb	O	hmastlock_ahb: RFU <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
hprot_ahb[3:0]	O	hprot_ahb: RFU <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
hsize_ahb[2:0]	O	hsize_ahb: RFU <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
htrans_ahb[1:0]	O	htrans_ahb: RFU <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
hwdata_ahb[31:0]	O	hwdata_ahb: RFU <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
hwrite_ahb	O	hwrite_ahb: RFU <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk

Port Name	I/O	Description
hclk_ahb	O	hclk_ahb: RFU <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
hresetn_ahb	O	hresetn_ahb: RFU <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
hrdata_ahb[31:0]	I	hrdata_ahb: RFU <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
hresp_ahb	I	hresp_ahb: RFU <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
hreadyout_ahb	I	hreadyout_ahb: RFU <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk

## 3.9 APB Signals

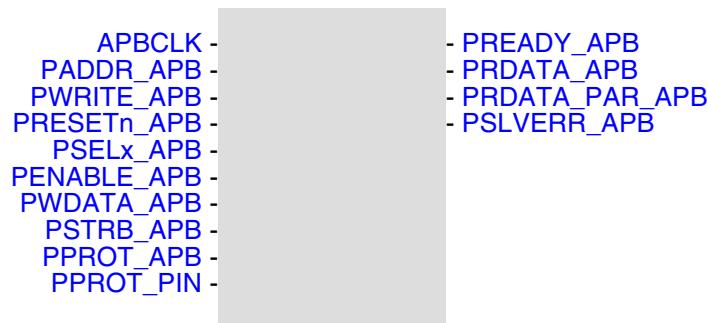


Table 3-9 APB Signals

Port Name	I/O	Description
APBCLK	I	clock for APB interface <b>Exists:</b> Always <b>Synchronous To:</b> None
PADDR_APB[31:0]	I	Interface z Address. This is the APB address bus. <b>Exists:</b> Always <b>Synchronous To:</b> APBCLK
PWRITE_APB	I	APB write access when HIGH and an APB read access when LOW <b>Exists:</b> Always <b>Synchronous To:</b> APBCLK
PRESETn_APB	I	APB Reset <b>Exists:</b> Always <b>Synchronous To:</b> APBCLK
PSELx_APB	I	indicates that the slave device is selected <b>Exists:</b> Always <b>Synchronous To:</b> APBCLK
PENABLE_APB	I	PENABLE_APB: signal indicates the second and subsequent cycles of an APB transfer <b>Exists:</b> Always <b>Synchronous To:</b> APBCLK
PWDATA_APB[31:0]	I	APB write data <b>Exists:</b> Always <b>Synchronous To:</b> APBCLK
PSTRB_APB[3:0]	I	APB write strobe <b>Exists:</b> Always <b>Synchronous To:</b> APBCLK

Port Name	I/O	Description
PPROT_APB[2:0]	I	APB protect type <b>Exists:</b> Always <b>Synchronous To:</b> APBCLK
PREADY_APB	O	APB ready <b>Exists:</b> Always <b>Synchronous To:</b> APBCLK
PRDATA_APB[31:0]	O	APB read data <b>Exists:</b> Always <b>Synchronous To:</b> APBCLK
PRDATA_PAR_APB[3:0]	O	APB read data parity <b>Exists:</b> Always <b>Synchronous To:</b> APBCLK
PSLVERR_APB	O	APB transfer failure <b>Exists:</b> Always <b>Synchronous To:</b> APBCLK
PPROT_PIN[2:0]	I	PPROT_PIN: signal indicates the second and subsequent cycles of an APB transfer <b>Exists:</b> Always <b>Synchronous To:</b> APBCLK

## 3.10 SRAM Signals

```
iccm0_data_dout0 - iccm0_bank0_clk  
iccm0_data_dout1 - iccm0_data_din0  
dccm_bank0_dout - iccm0_data_addr0  
dccm_bank1_dout - iccm0_data_cen0  
      bc_dout0 - iccm0_data_wen0  
      gs_dout0 - iccm0_data_wem0  
      bc_dout1 - iccm0_bank1_clk  
      gs_dout1 - iccm0_data_din1  
acsrm_data_dout - iccm0_data_addr1  
      pie_data_dout - iccm0_data_cen1  
                                - iccm0_data_wen1  
                                - iccm0_data_wem1  
                                - clk_dccm_bank0_lo  
                                - clk_dccm_bank0_hi  
                                - dccm_bank0_cs_lo  
                                - dccm_bank0_cs_hi  
                                - dccm_bank0_addr_lo  
                                - dccm_bank0_addr_hi  
                                - dccm_bank0_we_lo  
                                - dccm_bank0_we_hi  
                                - dccm_bank0_wem  
                                - dccm_bank0_din  
                                - clk_dccm_bank1_lo  
                                - clk_dccm_bank1_hi  
                                - dccm_bank1_cs_lo  
                                - dccm_bank1_cs_hi  
                                - dccm_bank1_addr_lo  
                                - dccm_bank1_addr_hi  
                                - dccm_bank1_we_lo  
                                - dccm_bank1_we_hi  
                                - dccm_bank1_wem  
                                - dccm_bank1_din  
      bc_din0 -  
      bc_addr0 -  
      bc_me0 -  
      bc_we0 -  
      bc_wem0 -  
      gs_din0 -  
      gs_addr0 -  
      gs_me0 -  
      gs_we0 -  
      gs_wem0 -  
      bc_ram0_clk -  
      gs_ram0_clk -  
      bc_din1 -  
      bc_addr1 -  
      bc_me1 -  
      bc_we1 -  
      bc_wem1 -  
      gs_din1 -  
      gs_addr1 -  
      gs_me1 -  
      gs_we1 -  
      gs_wem1 -  
      bc_ram1_clk -  
      gs_ram1_clk -  
      acsm_data_din -  
      acsm_data_addr -  
      acsm_data_ce -
```

- acsm\_data\_we  
 - acsm\_data\_clk  
 - pie\_data\_din  
 - pie\_data\_addr  
 - pie\_data\_ce  
 - pie\_data\_we  
 - pie\_data\_clk

**Table 3-10 SRAM Signals**

<b>Port Name</b>	<b>I/O</b>	<b>Description</b>
iccm0_bank0_clk	O	ICCM Bank0 memory clk <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
iccm0_data_dout0[77:0]	I	ICCM Bank0 memory Data output 0 <b>Exists:</b> Always <b>Synchronous To:</b> iccm0_bank0_clk
iccm0_data_din0[77:0]	O	ICCM Bank0 memory Data input 0 <b>Exists:</b> Always <b>Synchronous To:</b> iccm0_bank0_clk
iccm0_data_addr0[16:4]	O	ICCM Bank0 memory Data Addr 0 <b>Exists:</b> Always <b>Synchronous To:</b> iccm0_bank0_clk
iccm0_data_cen0	O	ICCM Bank0 memory Data cen0 <b>Exists:</b> Always <b>Synchronous To:</b> iccm0_bank0_clk
iccm0_data_wen0	O	ICCM Bank0 memory Data wen0 <b>Exists:</b> Always <b>Synchronous To:</b> iccm0_bank0_clk
iccm0_data_wem0[9:0]	O	ICCM Bank0 memory Data wem0 <b>Exists:</b> Always <b>Synchronous To:</b> iccm0_bank0_clk
iccm0_bank1_clk	O	ICCM Bank1 memory clk <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
iccm0_data_dout1[77:0]	I	ICCM Bank1 memory Data output 1 <b>Exists:</b> Always <b>Synchronous To:</b> iccm0_bank1_clk
iccm0_data_din1[77:0]	O	ICCM Bank1 memory Data input 1 <b>Exists:</b> Always <b>Synchronous To:</b> iccm0_bank1_clk

Port Name	I/O	Description
iccm0_data_addr1[16:4]	O	ICCM Bank1 memory Data Addr 1 <b>Exists:</b> Always <b>Synchronous To:</b> iccm0_bank1_clk
iccm0_data_cen1	O	ICCM Bank1 memory Data cen1 <b>Exists:</b> Always <b>Synchronous To:</b> iccm0_bank1_clk
iccm0_data_wen1	O	ICCM Bank1 memory Data wen1 <b>Exists:</b> Always <b>Synchronous To:</b> iccm0_bank1_clk
iccm0_data_wem1[9:0]	O	ICCM Bank1 memory Data wem1 <b>Exists:</b> Always <b>Synchronous To:</b> iccm0_bank1_clk
clk_dccm_bank0_lo	O	DCCM memory Bank 0 clk Low <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
clk_dccm_bank0_hi	O	DCCM memory Bank 0 clk High <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dccm_bank0_cs_lo	O	DCCM memory Bank 0 cs low <b>Exists:</b> Always <b>Synchronous To:</b> clk_dccm_bank0_lo
dccm_bank0_cs_hi	O	DCCM memory Bank 0 cs high <b>Exists:</b> Always <b>Synchronous To:</b> clk_dccm_bank0_hi
dccm_bank0_addr_lo[12:0]	O	DCCM memory Bank 0 addr low <b>Exists:</b> Always <b>Synchronous To:</b> clk_dccm_bank0_lo
dccm_bank0_addr_hi[12:0]	O	DCCM memory Bank 0 addr high <b>Exists:</b> Always <b>Synchronous To:</b> clk_dccm_bank0_hi
dccm_bank0_we_lo	O	DCCM memory Bank 0 we low <b>Exists:</b> Always <b>Synchronous To:</b> clk_dccm_bank0_lo
dccm_bank0_we_hi	O	DCCM memory Bank 0 we high <b>Exists:</b> Always <b>Synchronous To:</b> clk_dccm_bank0_hi

Port Name	I/O	Description
dccm_bank0_wem[9:0]	O	DCCM memory Bank 0 wem <b>Exists:</b> Always <b>Synchronous To:</b> clk_dccm_bank0_lo, clk_dccm_bank0_hi
dccm_bank0_din[77:0]	O	DCCM memory Bank 0 data in <b>Exists:</b> Always <b>Synchronous To:</b> clk_dccm_bank0_lo, clk_dccm_bank0_hi
dccm_bank0_dout[77:0]	I	DCCM memory Bank 0 data out <b>Exists:</b> Always <b>Synchronous To:</b> clk_dccm_bank0_lo, clk_dccm_bank0_hi
clk_dccm_bank1_lo	O	DCCM memory Bank 1 clk Low <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
clk_dccm_bank1_hi	O	DCCM memory Bank 1 clk High <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
dccm_bank1_cs_lo	O	DCCM memory Bank 1 cs low <b>Exists:</b> Always <b>Synchronous To:</b> clk_dccm_bank1_lo
dccm_bank1_cs_hi	O	DCCM memory Bank 1 cs high <b>Exists:</b> Always <b>Synchronous To:</b> clk_dccm_bank1_hi
dccm_bank1_addr_lo[12:0]	O	DCCM memory Bank 1 addr low <b>Exists:</b> Always <b>Synchronous To:</b> clk_dccm_bank1_lo
dccm_bank1_addr_hi[12:0]	O	DCCM memory Bank 1 addr high <b>Exists:</b> Always <b>Synchronous To:</b> clk_dccm_bank1_hi
dccm_bank1_we_lo	O	DCCM memory Bank 1 we low <b>Exists:</b> Always <b>Synchronous To:</b> clk_dccm_bank1_lo
dccm_bank1_we_hi	O	DCCM memory Bank 1 we high <b>Exists:</b> Always <b>Synchronous To:</b> clk_dccm_bank1_hi
dccm_bank1_wem[9:0]	O	DCCM memory Bank 1 wem <b>Exists:</b> Always <b>Synchronous To:</b> clk_dccm_bank1_lo, clk_dccm_bank1_hi

Port Name	I/O	Description
dccm_bank1_din[77:0]	O	DCCM memory Bank 1 data in <b>Exists:</b> Always <b>Synchronous To:</b> clk_dccm_bank1_lo, clk_dccm_bank1_hi
dccm_bank1_dout[77:0]	I	DCCM memory Bank 1 data out <b>Exists:</b> Always <b>Synchronous To:</b> clk_dccm_bank1_lo, clk_dccm_bank1_hi
bc_din0[63:0]	O	Branch Cache memory BC data in 0 <b>Exists:</b> Always <b>Synchronous To:</b> bc_ram0_clk
bc_addr0[11:4]	O	Branch Cache memory BC Addr 0 <b>Exists:</b> Always <b>Synchronous To:</b> bc_ram0_clk
bc_me0	O	Branch Cache memory BC me 0 <b>Exists:</b> Always <b>Synchronous To:</b> bc_ram0_clk
bc_we0	O	Branch Cache memory BC we 0 <b>Exists:</b> Always <b>Synchronous To:</b> bc_ram0_clk
bc_wem0[63:0]	O	Branch Cache memory BC wem 0 <b>Exists:</b> Always <b>Synchronous To:</b> bc_ram0_clk
bc_dout0[63:0]	I	Branch Cache memory BC data out 0 <b>Exists:</b> Always <b>Synchronous To:</b> bc_ram0_clk
gs_din0[7:0]	O	Predictive Table memory GS data in 0 <b>Exists:</b> Always <b>Synchronous To:</b> gs_ram0_clk
gs_addr0[13:4]	O	Predictive Table memory GS Addr 0 <b>Exists:</b> Always <b>Synchronous To:</b> gs_ram0_clk
gs_me0	O	Predictive Table memory GS me 0 <b>Exists:</b> Always <b>Synchronous To:</b> gs_ram0_clk
gs_we0	O	Predictive Table memory GS we 0 <b>Exists:</b> Always <b>Synchronous To:</b> gs_ram0_clk

Port Name	I/O	Description
gs_wem0[7:0]	O	Predictive Table memory GS wem 0 <b>Exists:</b> Always <b>Synchronous To:</b> gs_ram0_clk
gs_dout0[7:0]	I	Predictive Table memory GS data out 0 <b>Exists:</b> Always <b>Synchronous To:</b> gs_ram0_clk
bc_ram0_clk	O	Branch Cache memory BC RAM clk 0 <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
gs_ram0_clk	O	Predictive Table memory GS RAM clk 0 <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
bc_din1[63:0]	O	Branch Cache memory BC data in 1 <b>Exists:</b> Always <b>Synchronous To:</b> bc_ram1_clk
bc_addr1[11:4]	O	Branch Cache memory BC Addr 1 <b>Exists:</b> Always <b>Synchronous To:</b> bc_ram1_clk
bc_me1	O	Branch Cache memory BC me 1 <b>Exists:</b> Always <b>Synchronous To:</b> bc_ram1_clk
bc_we1	O	Branch Cache memory BC we 1 <b>Exists:</b> Always <b>Synchronous To:</b> bc_ram1_clk
bc_wem1[63:0]	O	Branch Cache memory BC wem 1 <b>Exists:</b> Always <b>Synchronous To:</b> bc_ram1_clk
bc_dout1[63:0]	I	Branch Cache memory BC data out 1 <b>Exists:</b> Always <b>Synchronous To:</b> bc_ram1_clk
gs_din1[7:0]	O	Predictive Table memory GS data in 1 <b>Exists:</b> Always <b>Synchronous To:</b> gs_ram1_clk
gs_addr1[13:4]	O	Predictive Table memory GS Addr 1 <b>Exists:</b> Always <b>Synchronous To:</b> gs_ram1_clk

Port Name	I/O	Description
gs_me1	O	Predictive Table memory GS me 1 <b>Exists:</b> Always <b>Synchronous To:</b> gs_ram1_clk
gs_we1	O	Predictive Table memory GS we 1 <b>Exists:</b> Always <b>Synchronous To:</b> gs_ram1_clk
gs_wem1[7:0]	O	Predictive Table memory GS wem 1 <b>Exists:</b> Always <b>Synchronous To:</b> gs_ram1_clk
gs_dout1[7:0]	I	Predictive Table memory GS data out 1 <b>Exists:</b> Always <b>Synchronous To:</b> gs_ram1_clk
bc_ram1_clk	O	Branch Cache memory BC RAM clk 1 <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
gs_ram1_clk	O	Predictive Table memory GS RAM clk 1 <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk
acsm_data_dout[71:0]	I	ACSM memory Read Data <b>Exists:</b> Always <b>Synchronous To:</b> acsm_data_clk
acsm_data_din[71:0]	O	ACSM memory Write Data <b>Exists:</b> Always <b>Synchronous To:</b> acsm_data_clk
acsm_data_addr[10:0]	O	ACSM memory Address <b>Exists:</b> Always <b>Synchronous To:</b> acsm_data_clk
acsm_data_ce	O	ACSM memory Chip Enable <b>Exists:</b> Always <b>Synchronous To:</b> acsm_data_clk
acsm_data_we	O	ACSM memory Write Enable <b>Exists:</b> Always <b>Synchronous To:</b> acsm_data_clk
acsm_data_clk	O	ACSM memory Clk <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk

Port Name	I/O	Description
pie_data_dout[55:0]	I	PIE memory Read Data <b>Exists:</b> Always <b>Synchronous To:</b> pie_data_clk
pie_data_din[55:0]	O	PIE memory Write Data <b>Exists:</b> Always <b>Synchronous To:</b> pie_data_clk
pie_data_addr[11:0]	O	PIE memory Address. Bit [11] is RFU and should be left dangling. <b>Exists:</b> Always <b>Synchronous To:</b> pie_data_clk
pie_data_ce	O	PIE memory Chip Enable <b>Exists:</b> Always <b>Synchronous To:</b> pie_data_clk
pie_data_we	O	PIE memory Write Enable <b>Exists:</b> Always <b>Synchronous To:</b> pie_data_clk
pie_data_clk	O	PIE memory clock <b>Exists:</b> Always <b>Synchronous To:</b> DfiClk

### 3.11 Bypass mode Signals

TxBypassMode_MEMRESET_L -	- RxBypassDataPad_MEMRESET_L
TxBypassData_MEMRESET_L -	- RxBypassDataPad.DTO0
RxBypassPadEn_MEMRESET_L -	- RxBypassDataRcv.DTO0
TxBypassMode.DTO0 -	- RxBypassDataPad.DTO1
TxBypassOE.DTO0 -	- RxBypassDataRcv.DTO1
TxBypassData.DTO0 -	- RxBypassDataPad.Bz.Dy (for z = 0; z <= 3)(for y = 0; y <= 12)
RxBypassPadEn.DTO0 -	- RxBypassDataRcv.Bz.Dy (for z = 0; z <= 3)(for y = 0; y <= 12)
RxBypassRcvEn.DTO0 -	- RxBypassDataRcv.Ay (for y = 0; y <= 19)
TxBypassMode.DTO1 -	- RxBypassDataPad.A
TxBypassOE.DTO1 -	- RxBypassDataRcv.CKz.T (for z = 0; z <= 1)
TxBypassData.DTO1 -	- RxBypassDataRcv.CKz.C (for z = 0; z <= 1)
RxBypassPadEn.DTO1 -	- RxBypassDataPad.CKz.T (for z = 0; z <= 1)
RxBypassRcvEn.DTO1 -	- RxBypassDataPad.CKz.C (for z = 0; z <= 1)
TxBypassData_A -	
TxBypassData.CKz.T (for z = 0; z <= 1) -	
TxBypassData.CKz.C (for z = 0; z <= 1) -	
TxBypassMode_A -	
TxBypassMode.CKz.T (for z = 0; z <= 1) -	
TxBypassMode.CKz.C (for z = 0; z <= 1) -	
TxBypassOE_A -	
TxBypassOE.CKz.T (for z = 0; z <= 1) -	
TxBypassOE.CKz.C (for z = 0; z <= 1) -	
RxBypassPadEn_A -	
RxBypassPadEn.CKz (for z = 0; z <= 1) -	
RxBypassRcvEn_A -	
RxBypassRcvEn.CKz (for z = 0; z <= 1) -	
TxBypassMode.Bz.Dy (for z = 0; z <= 3)(for y = 0; y <= 12) -	
TxBypassOE.Bz.Dy (for z = 0; z <= 3)(for y = 0; y <= 12) -	
TxBypassData.Bz.Dy (for z = 0; z <= 3)(for y = 0; y <= 12) -	
RxBypassRcvEn.Bz.Dy (for z = 0; z <= 3)(for y = 0; y <= 12) -	
RxBypassPadEn.Bz.Dy (for z = 0; z <= 3)(for y = 0; y <= 12) -	

Table 3-11 Bypass mode Signals

Port Name	I/O	Description
TxBypassMode_MEMRESET_L	I	async tx bypass path enable for MEMRESET_L pad <b>Exists:</b> Always <b>Synchronous To:</b> None
TxBypassData_MEMRESET_L	I	async tx data for MEMRESET_L pad in bypass mode <b>Exists:</b> Always <b>Synchronous To:</b> None

Port Name	I/O	Description
RxBypassPadEn_MEMRESET_L	I	bypass pad enable for MEMRESET_L pad <b>Exists:</b> Always <b>Synchronous To:</b> None
RxBypassDataPad_MEMRESET_L	O	async data sampled from MEMRESET_L pad for bypass mode <b>Exists:</b> Always <b>Synchronous To:</b> None
TxBypassMode.DTO0	I	tx bypass path enable for DTO pad. Only exist when DWC_LPDDR5XPHY.DTO_ENABLED is defined <b>Exists:</b> Always <b>Synchronous To:</b> None
TxBypassOE.DTO0	I	async OE for DTO pad in bypass mode. Only exist when DWC_LPDDR5XPHY.DTO_ENABLED is defined <b>Exists:</b> Always <b>Synchronous To:</b> None
TxBypassData.DTO0	I	async tx data for DTO pad in bypass mode. Only exist when DWC_LPDDR5XPHY.DTO_ENABLED is defined <b>Exists:</b> Always <b>Synchronous To:</b> None
RxBypassPadEn.DTO0	I	rx bypass pad enable for DTO pad. Only exist when DWC_LPDDR5XPHY.DTO_ENABLED is defined <b>Exists:</b> Always <b>Synchronous To:</b> None
RxBypassRcvEn.DTO0	I	Receiver-RX bypass path enable for DTO pad. Only exist when DWC_LPDDR5XPHY.DTO_ENABLED is defined <b>Exists:</b> Always <b>Synchronous To:</b> None
RxBypassDataPad.DTO0	O	async data sampled from DTO pad for bypass mode. Only exist when DWC_LPDDR5XPHY.DTO_ENABLED is defined <b>Exists:</b> Always <b>Synchronous To:</b> None
RxBypassDataRcv.DTO0[1:0]	O	asynchronous data sampled from DTO receiver for bypass mode. Only exist when DWC_LPDDR5XPHY.DTO_ENABLED is defined <b>Exists:</b> Always <b>Synchronous To:</b> Pclk
TxBypassMode.DTO1	I	tx bypass path enable for DTO pad. Only exist when DWC_LPDDR5XPHY.DTO_ENABLED is defined <b>Exists:</b> Always <b>Synchronous To:</b> None

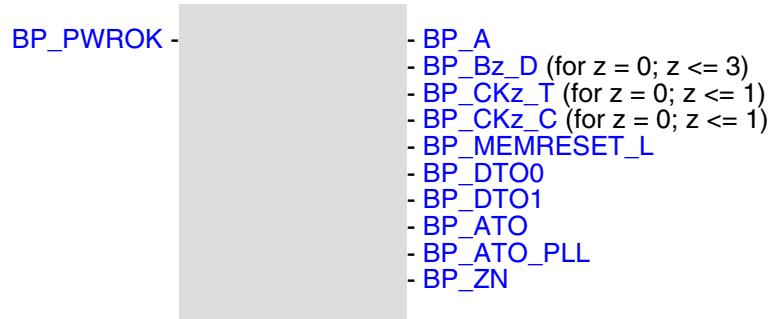
Port Name	I/O	Description
TxBypassOE_DTO1	I	<p>async OE for DTO pad in bypass mode. Only exist when DWC_LPDDR5XPHY.DTO_ENABLED is defined</p> <p><b>Exists:</b> Always</p> <p><b>Synchronous To:</b> None</p>
TxBypassData_DTO1	I	<p>async tx data for DTO pad in bypass mode. Only exist when DWC_LPDDR5XPHY.DTO_ENABLED is defined</p> <p><b>Exists:</b> Always</p> <p><b>Synchronous To:</b> None</p>
RxBypassPadEn_DTO1	I	<p>rx bypass pad enable for DTO pad. Only exist when DWC_LPDDR5XPHY.DTO_ENABLED is defined</p> <p><b>Exists:</b> Always</p> <p><b>Synchronous To:</b> None</p>
RxBypassRcvEn_DTO1	I	<p>Receiver-RX bypass path enable for DTO pad. Only exist when DWC_LPDDR5XPHY.DTO_ENABLED is defined</p> <p><b>Exists:</b> Always</p> <p><b>Synchronous To:</b> None</p>
RxBypassDataPad_DTO1	O	<p>async data sampled from DTO pad for bypass mode. Only exist when DWC_LPDDR5XPHY.DTO_ENABLED is defined</p> <p><b>Exists:</b> Always</p> <p><b>Synchronous To:</b> None</p>
RxBypassDataRcv_DTO1[1:0]	O	<p>asynchronous data sampled from DTO receiver for bypass mode. Only exist when DWC_LPDDR5XPHY.DTO_ENABLED is defined</p> <p><b>Exists:</b> Always</p> <p><b>Synchronous To:</b> Pclk</p>
TxBypassData_A[((10*DWC_LPDDR5XP HY_NUM_CHANNELS)-1):0]	I	<p>async TX data for addr/command pad for bypass mode</p> <p><b>Exists:</b> Always</p> <p><b>Synchronous To:</b> None</p>
TxBypassData_CKz_T (for z = 0; z <= 1)	I	<p>TxBypassData_CK0_T: async TX data for CK pad in bypass mode</p> <p><b>Exists:</b> Always</p> <p><b>Synchronous To:</b> None</p>
TxBypassData_CKz_C (for z = 0; z <= 1)	I	<p>TxBypassData_CK0_C: async TX data for CK pad in bypass mode</p> <p><b>Exists:</b> Always</p> <p><b>Synchronous To:</b> None</p>
TxBypassMode_A[((10*DWC_LPDDR5XP HY_NUM_CHANNELS)-1):0]	I	<p>async TX path enable for addr/command pad in bypass mode</p> <p><b>Exists:</b> Always</p> <p><b>Synchronous To:</b> None</p>

Port Name	I/O	Description
TxBypassMode_CKz_T (for z = 0; z <= 1)	I	TxBypassMode_CK0_T: async TX path enable for CK pad in bypass mode <b>Exists:</b> Always <b>Synchronous To:</b> None
TxBypassMode_CKz_C (for z = 0; z <= 1)	I	TxBypassMode_CK0_C: async TX path enable for CK pad in bypass mode <b>Exists:</b> Always <b>Synchronous To:</b> None
TxBypassOE_A[((10*DWC_LPDDR5XPHY_NUM_CHANNELS)-1):0]	I	async OE for addr/command pad for bypass mode <b>Exists:</b> Always <b>Synchronous To:</b> None
TxBypassOE_CKz_T (for z = 0; z <= 1)	I	TxBypassOE_CK0_T-async OE for CK pad in bypass mode <b>Exists:</b> Always <b>Synchronous To:</b> None
TxBypassOE_CKz_C (for z = 0; z <= 1)	I	TxBypassOE_CK0_C-async OE for CK pad in bypass mode <b>Exists:</b> Always <b>Synchronous To:</b> None
RxBypassPadEn_A[((10*DWC_LPDDR5XPHY_NUM_CHANNELS)-1):0]	I	per lane CMOS RX bypass path enable for each CA pad <b>Exists:</b> Always <b>Synchronous To:</b> None
RxBypassPadEn_CKz (for z = 0; z <= 1)	I	RxBypassPadEn_CK0: CMOS-RX bypass path enable for CK true and complement pad <b>Exists:</b> Always <b>Synchronous To:</b> None
RxBypassRcvEn_A[((10*DWC_LPDDR5XPHY_NUM_CHANNELS)-1):0]	I	per lane receive RX bypass path enable for addr/command pad <b>Exists:</b> Always <b>Synchronous To:</b> None
RxBypassRcvEn_CKz (for z = 0; z <= 1)	I	RxBypassRcvEn_CK0: Receiver-RX bypass path enable for CK true and complement pad <b>Exists:</b> Always <b>Synchronous To:</b> None
TxBypassMode_Bz_Dy (for z = 0; z <= 3)(for y = 0; y <= 12)	I	async tx path enable for data+strobe pads for bypass mode <b>Exists:</b> Always <b>Synchronous To:</b> None
TxBypassOE_Bz_Dy (for z = 0; z <= 3)(for y = 0; y <= 12)	I	async OE for data+strobe pads for bypass mode <b>Exists:</b> Always <b>Synchronous To:</b> None

Port Name	I/O	Description
TxBypassData_Bz_Dy (for z = 0; z <= 3)(for y = 0; y <= 12)	I	async tx data for data+strobe pads for bypass mode <b>Exists:</b> Always <b>Synchronous To:</b> None
RxBypassRcvEn_Bz_Dy (for z = 0; z <= 3)(for y = 0; y <= 12)	I	per lane receive RX bypass path enable for data lane <b>Exists:</b> Always <b>Synchronous To:</b> None
RxBypassPadEn_Bz_Dy (for z = 0; z <= 3)(for y = 0; y <= 12)	I	per lane CMOS RX bypass path enable for data lane <b>Exists:</b> Always <b>Synchronous To:</b> None
RxBypassDataPad_Bz_Dy (for z = 0; z <= 3)(for y = 0; y <= 12)	O	per lane asynchronous data sampled from each DQ pad for bypass mode <b>Exists:</b> Always <b>Synchronous To:</b> None
RxBypassDataRcv_Bz_Dy[1:0] (for z = 0; z <= 3)(for y = 0; y <= 12)	O	per lane asynchronous data sampled from each DQ receiver for bypass mode <b>Exists:</b> Always <b>Synchronous To:</b> Pclk
RxBypassDataRcv_Ay[1:0] (for y = 0; y <= 19)	O	per lane asynchronous data sampled from each CA pad for bypass mode <b>Exists:</b> Always <b>Synchronous To:</b> None
RxBypassDataPad_A[((10*DWC_LPDDR5_XPHY_NUM_CHANNELS)-1):0]	O	per lane asynchronous data sampled from each CA receiver for bypass mode <b>Exists:</b> Always <b>Synchronous To:</b> None
RxBypassDataRcv_CKz_T[1:0] (for z = 0; z <= 1)	O	RxBypassDataRcv_CK0_T-async data sampled from CK True receiver for bypass mode <b>Exists:</b> Always <b>Synchronous To:</b> None
RxBypassDataRcv_CKz_C[1:0] (for z = 0; z <= 1)	O	RxBypassDataRcv_CK0_C: async data sampled from CK Complement receiver for bypass mode <b>Exists:</b> Always <b>Synchronous To:</b> None
RxBypassDataPad_CKz_T (for z = 0; z <= 1)	O	RxBypassDataPad_CK0_T: async data sampled from CK True for bypass mode <b>Exists:</b> Always <b>Synchronous To:</b> None

Port Name	I/O	Description
RxBypassDataPad_CKz_C (for z = 0; z <= 1)	O	RxBypassDataPad_CK0_C: async data sampled from CK Complement for bypass mode <b>Exists:</b> Always <b>Synchronous To:</b> None

## 3.12 Bump Signals

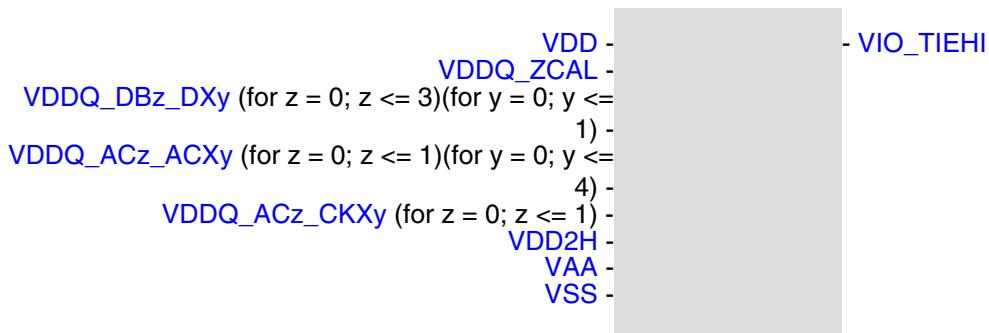


**Table 3-12 Bump Signals**

Port Name	I/O	Description
BP_A[((10*DWC_LPDDR5XPHY_NUM_C HANNELS)-1):0]	IO	DRAM address and command bits <b>Exists:</b> Always <b>Synchronous To:</b> None
BP_Bz_D[12:0] (for z = 0; z <= 3)	IO	BP_B0_D: DRAM data bits and strobes <b>Exists:</b> Always <b>Synchronous To:</b> None
BP_CKz_T (for z = 0; z <= 1)	IO	BP_CK0_T: DRAM clock <b>Exists:</b> Always <b>Synchronous To:</b> None
BP_CKz_C (for z = 0; z <= 1)	IO	BP_CK0_C: DRAM clock (complement) <b>Exists:</b> Always <b>Synchronous To:</b> None
BP_PWROK	I	Retention mode control <b>Exists:</b> Always <b>Synchronous To:</b> None
BP_MEMRESET_L	O	DRAM reset <b>Exists:</b> Always <b>Synchronous To:</b> None
BP.DTO0	IO	Digital test output. Only exist when DWC_LPDDR5XPHY.DTO_ENABLED is defined <b>Exists:</b> Always <b>Synchronous To:</b> None
BP.DTO1	IO	Digital test output. Only exist when DWC_LPDDR5XPHY.DTO_ENABLED is defined <b>Exists:</b> Always <b>Synchronous To:</b> None

Port Name	I/O	Description
BP_ATO	IO	Analog test output <b>Exists:</b> Always <b>Synchronous To:</b> None
BP_ATO_PLL	IO	Analog test output from PLL <b>Exists:</b> Always <b>Synchronous To:</b> None
BP_ZN	IO	calibration external reference resistor <b>Exists:</b> Always <b>Synchronous To:</b> None

### 3.13 Power and Ground Signals



**Table 3-13 Power and Ground Signals**

Port Name	I/O	Description
VDD	I	digital logic power supply <b>Exists:</b> Always <b>Synchronous To:</b> None
VDDQ_ZCAL	I	ZCAL IO power supply <b>Exists:</b> Always <b>Synchronous To:</b> None
VDDQ_DBz_DXy (for z = 0; z <= 3)(for y = 0; y <= 1)	I	VDDQ_DB0_DX0: DB0 DX0 IO power supply <b>Exists:</b> Always <b>Synchronous To:</b> None
VDDQ_ACz_ACXy (for z = 0; z <= 1)(for y = 0; y <= 4)	I	VDDQ_AC0_ACX0: AC0 ACX0 IO power supply (VDDQ_AC0_ACX4 Only exists when DWC_LPDDR5XPHY.DTO_ENABLED is defined) <b>Exists:</b> Always <b>Synchronous To:</b> None
VDDQ_ACz_CKXy (for z = 0; z <= 1)	I	VDDQ_AC0_CKX0: AC0 CKX0 IO power supply <b>Exists:</b> Always <b>Synchronous To:</b> None
VDD2H	I	CMOS output power supply <b>Exists:</b> Always <b>Synchronous To:</b> None
VAA	I	PLL power supply <b>Exists:</b> Always <b>Synchronous To:</b> None
VSS	I	Ground <b>Exists:</b> Always <b>Synchronous To:</b> None

Port Name	I/O	Description
VIO_TIEHI	O	TIEHI output in VDD2H domain for driving BP_PWROK if retention is not supported <b>Exists:</b> Always <b>Synchronous To:</b> None

# 4

# Signal Mapping

---

The following sections are included in this chapter:

- “DFI Pin List and Pin Mapping” on page [114](#)
- “DFI Pin Swizzling” on page [122](#)
- “DFI Interface Implementation Details” on page [123](#)
- “DFI Phase Bits, Chip Selects and Data Bits” on page [124](#)
- “DFI Command Phase Timing” on page [125](#)

## 4.1 DFI Pin List and Pin Mapping

### 4.1.1 DBYTE Assignment

CH\_A : 0,1  
CH\_B : 2,3

### 4.1.2 DFI Pin List

Note:

NUM\_ADDR = 14  
NUM\_RANKS = 2  
NUM\_DBYTES = 4  
NUM\_DQ = 32

### 4.1.3 LPDDR5/5X DFI-to-SDRAM Pin Map DWC\_LPDDR5XPHY\_NUM\_CHANNELS\_1

Table 4-1 LPDDR5/5X DFI-to-SDRAM Pin Map (Single Channel)

DFI Signal	RTL Bump Name	LPDDR5 (CH A)	LPDDR5 (CH B)
dfi0_address_P0[7,0]	BP_A[0]	CAA[0]	
dfi0_address_P0[8,1]	BP_A[1]	CAA[1]	
dfi0_address_P0[9,2]	BP_A[2]	CAA[2]	
dfi0_address_P0[10,3]	BP_A[3]	CAA[3]	
dfi0_address_P0[11,4]	BP_A[6]	CAA[4]	
dfi0_address_P0[12,5]	BP_A[7]	CAA[5]	
dfi0_address_P0[13,6]	BP_A[8]	CAA[6]	
dfi0_cs_P0[0]	BP_A[4]	CSA[0]	
dfi0_cs_P0[1]	BP_A[5]	CSA[1]	
dfi0_dram_clk_disable_P0[0]	BP_CK0_T	CLKA_T	
dfi0_dram_clk_disable_P0[0]	BP_CK0_C	CLKA_C	
dfi0_reset_n	BP_MEMRESET_L	RESET_N	
dfi0_wrdata_P{0..3}/ dfi0_rddata_W{0..3}[{23:16,7:0}], dfi0_wrdata_cs_P{0..3}[1:0], dfi0_rddata_cs_P{0..3}[1:0],	BP_B0_D[7:0]	DQA[7:0]	
dfi0_rddata_en_P{0..3}[0]	BP_B0_D[8]	DQSA_C[0]	

DFI Signal	RTL Bump Name	LPDDR5 (CH A)	LPDDR5 (CH B)
dfi0_wrdata_link_ecc_P{0..3}[2,0], dfi0_rddata_en_P{0..3}[0]	BP_B0_D[9]	DPARA[0] or DQSA_T[0]	
dfi0_wck_en_P{0..3}[0], dfi0_wck_cs_P{0..3}[1 :0]	BP_B0_D[10]	WCKA_C[0]	
dfi0_wck_en_P{0..3}[0], dfi0_wck_cs_P{0..3}[1 :0]	BP_B0_D[11]	WCKA_T[0]	
dfi0_wrdata_mask_P{0..3}[2,0]/ dfi0_rddata_dbi_W{0..3}[2,0],	BP_B0_D[12]	DM/DBIA[0]	
dfi0_wrdata_P{0..3}/ dfi0_rddata_W{0..3}[31:24,15:8], dfi0_wrdata_cs_P{0..3}[3:2], dfi0_rddata_cs_P{0..3}[3:2],	BP_B1_D[7:0]	DQA[15:8]	
dfi0_rddata_en_P{0..3}[1]	BP_B1_D[8]	DQSA_C[1]	
dfi0_wrdata_link_ecc_P{0..3}[3,1], dfi0_rddata_en_P{0..3}[1]	BP_B1_D[9]	DPARA[1] or DQSA_T[1]	
dfi0_wck_en_P{0..3}[1], dfi0_wck_cs_P{0..3}[3 :2]	BP_B1_D[10]	WCKA_C[1]	
dfi0_wck_en_P{0..3}[1], dfi0_wck_cs_P{0..3}[3 :2]	BP_B1_D[11]	WCKA_T[1]	
dfi0_wrdata_mask_P{0..3}[3,1]/ dfi0_rddata_dbi_W{0..3}[3,1],	BP_B1_D[12]	DM/DBIA[1]	

#### 4.1.4 LPDDR5/5X DFI-to-SDRAM Pin Map DWC\_LPDDR5XPHY\_NUM\_CHANNEL\_2

Table 4-2 LPDDR5/5X DFI-to-SDRAM Pin Map (Dual Channel, Two Ranks)

DFI Signal	RTL Bump Name	LPDDR5 (CH A)	LPDDR5 (CH B)
dfi0_address_P0[7,0]	BP_A[0]	CAA[0]	
dfi0_address_P0[8,1]	BP_A[1]	CAA[1]	
dfi0_address_P0[9,2]	BP_A[2]	CAA[2]	
dfi0_address_P0[10,3]	BP_A[3]	CAA[3]	
dfi0_address_P0[11,4]	BP_A[6]	CAA[4]	
dfi0_address_P0[12,5]	BP_A[7]	CAA[5]	
dfi0_address_P0[13,6]	BP_A[8]	CAA[6]	
dfi0_cs_P0[0]	BP_A[4]	CSA[0]	

<b>DFI Signal</b>	<b>RTL Bump Name</b>	<b>LPDDR5 (CH A)</b>	<b>LPDDR5 (CH B)</b>
dfi0_cs_P0[1]	BP_A[5]	CSA[1]	
dfi0_dram_clk_disable_P0[0]	BP_CK0_T	CLKA_T	
dfi0_dram_clk_disable_P0[0]	BP_CK0_C	CLKA_C	
dfi1_address_P0[7,0]	BP_A[10]		CAB[0]
dfi1_address_P0[8,1]	BP_A[11]		CAB[1]
dfi1_address_P0[9,2]	BP_A[12]		CAB[2]
dfi1_address_P0[10,3]	BP_A[13]		CAB[3]
dfi1_address_P0[11,4]	BP_A[16]		CAB[4]
dfi1_address_P0[12,5]	BP_A[17]		CAB[5]
dfi1_address_P0[13,6]	BP_A[18]		CAB[6]
dfi1_cs_P0[0]	BP_A[14]		CSB[0]
dfi1_cs_P0[1]	BP_A[15]		CSB[1]
dfi1_dram_clk_disable_P0[0]	BP_CK1_T		CLKB_T
dfi1_dram_clk_disable_P0[0]	BP_CK1_C		CLKB_C
dfi0_reset_n	BP_MEMRESET_L	RESET_N	
dfi0_wrdata_P{0..3}/ dfi0_rddata_W{0..3}[{23:16,7:0}], dfi0_wrdata_cs_P{0..3}[1:0], dfi0_rddata_cs_P{0..3}[1:0],	BP_B0_D[7:0]	DQA[7:0]	
dfi0_rddata_en_P{0..3}[0]	BP_B0_D[8]	DQSA_C[0]	
dfi0_wrdata_link_ecc_P{0..3}[2,0], dfi0_rddata_en_P{0..3}[0]	BP_B0_D[9]	DPARA[0] or DQSA_T[0]	
dfi0_wck_en_P{0..3}[0] , dfi0_wck_cs_P{0..3}[1 :0]	BP_B0_D[10]	WCKA_C[0]	
dfi0_wck_en_P{0..3}[0], dfi0_wck_cs_P{0..3}[1 :0]	BP_B0_D[11]	WCKA_T[0]	
dfi0_wrdata_mask_P{0..3}[2,0]/ dfi0_rddata_dbi_W{0..3}[2,0] ,	BP_B0_D[12]	DM/DBIA[0]	
dfi0_wrdata_P{0..3}/ dfi0_rddata_W{0..3}[31:24,15:8] , dfi0_wrdata_cs_P{0..3}[3:2], dfi0_rddata_cs_P{0..3}[3:2],	BP_B1_D[7:0]	DQA[15:8]	

DFI Signal	RTL Bump Name	LPDDR5 (CH A)	LPDDR5 (CH B)
dfi0_rddata_en_P{0..3}[1]	BP_B1_D[8]	DQSA_C[1]	
dfi0_wrdata_link_ecc_P{0..3}[3,1], dfi0_rddata_en_P{0..3}[1]	BP_B1_D[9]	DPARA[1] or DQSA_T[1]	
dfi0_wck_en_P{0..3}[1], dfi0_wck_cs_P{0..3}[3:2]	BP_B1_D[10]	WCKA_C[1]	
dfi0_wck_en_P{0..3}[1], dfi0_wck_cs_P{0..3}[3:2]	BP_B1_D[11]	WCKA_T[1]	
dfi0_wrdata_mask_P{0..3}[3,1]/ dfi0_rddata_dbi_W{0..3}[3,1],	BP_B1_D[12]	DM/DBIA[1]	
dfi1_wrdata_P{0..3}/ dfi1_rddata_W{0..3}[{23:16,7:0}], dfi1_wrdata_cs_P{0..3}[1:0], dfi1_rddata_cs_P{0..3}[1:0],	BP_B2_D[7:0]		DQB[7:0]
dfi1_rddata_en_P{0..3}[0]	BP_B2_D[8]		DQSB_C[0]
dfi1_wrdata_link_ecc_P{0..3}[2,0], dfi1_rddata_en_P{0..3}[0]	BP_B2_D[9]		DPARB[0] or DQSB_T[0]
dfi1_wck_en_P{0..3}[0], dfi1_wck_cs_P{0..3}[1:0]	BP_B2_D[10]		WCKB_C[0]
dfi1_wck_en_P{0..3}[0], dfi1_wck_cs_P{0..3}[1:0]	BP_B2_D[11]		WCKB_T[0]
dfi1_wrdata_mask_P{0..3}[2,0]/ dfi1_rddata_dbi_W{0..3}[2,0],	BP_B2_D[12]		DM/DBIB[0]
dfi1_wrdata_P{0..3}/ dfi1_rddata_W{0..3}[{31_24, 15:8}], dfi1_wrdata_cs_P{0..3}[3:2], dfi1_rddata_cs_P{0..3}[3:2],	BP_B3_D[7:0]		DQB[15:8]
dfi1_rddata_en_P{0..3}[1]	BP_B3_D[8]		DQSB_C[1]
dfi1_wrdata_link_ecc_P{0..3}[3,1], dfi1_rddata_en_P{0..3}[1]	BP_B3_D[9]		DPARB[1] or DQSB_T[1]
dfi1_wck_en_P{0..3}[1], dfi1_wck_cs_P{0..3}[3:2]	BP_B3_D[10]		WCKB_C[1]
dfi1_wck_en_P{0..3}[1], dfi1_wck_cs_P{0..3}[3:2]	BP_B3_D[11]		WCKB_T[1]
dfi1_wrdata_mask_P{0..3}[3,1]/ dfi1_rddata_dbi_W{0..3}[3,1],	BP_B3_D[12]		DM/DBIB[1]

#### 4.1.5 LPDDR4X DFI-to-SDRAM Pin Map DWC\_LPDDR5XPHY\_NUM\_CHANNEL\_1

**Table 4-3 LPDDR4X DFI-to-SDRAM Pin Map (Single Channel, Dual Rank)**

DFI Signal	RTL Bump Name	LPDDR4X (CH A)	LPDDR4X (CH B)
dfi0_address_P{0..3}[0]	BP_A[0]	CAA[0]	
dfi0_address_P{0..3}[1]	BP_A[1]	CAA[1]	
dfi0_address_P{0..3}[2]	BP_A[2]	CAA[2]	
dfi0_address_P{0..3}[3]	BP_A[3]	CAA[3]	
dfi0_address_P{0..3}[4]	BP_A[6]	CAA[4]	
dfi0_address_P{0..3}[5]	BP_A[7]	CAA[5]	
dfi0_cke_P{0..3}[0]	BP_A[4]	CKEA[0]	
dfi0_cke_P{0..3}[1]	BP_A[5]	CKEA[1]	
dfi0_cs_P{0..3}[0]	BP_A[8]	CSA[0]	
dfi0_cs_P{0..3}[1]	BP_A[9]	CSA[1]	
dfi0_dram_clk_disable_P{0..3}[0]	BP_CK0_T	CLKA_T	
dfi0_dram_clk_disable_P{0..3}[0]	BP_CK0_C	CLKA_C	
dfi_reset_n	BP_MEMRESET_L	RESET_N	
dfi0_wrdata_P{0..3}/ dfi0_rddata_W{0..3}[{23:16,7:0}], dfi0_wrdata_cs_P{0..3}[1:0], dfi0_rddata_cs_P{0..3}[1:0],	BP_B0_D[7:0]	DQA[7:0]	
dfi0_rddata_en_P{0..3}[0]	BP_B0_D[8]	DQSA_C[0]	
dfi0_rddata_en_P{0..3}[0]	BP_B0_D[9]	DPARA[0] or DQSA_T[0]	
	BP_B0_D[10]	Unused	
	BP_B0_D[11]	Unused	
dfi0_wrdata_mask_P{0..3}[2,0]/ dfi0_rddata_dbi_W{0..3}[2,0],	BP_B0_D[12]	DM/DBIA[0]	
dfi0_wrdata_P{0..3}/ dfi0_rddata_W{0..3}[31:24,15:8], dfi0_wrdata_cs_P{0..3}[3:2], dfi0_rddata_cs_P{0..3}[3:2],	BP_B1_D[7:0]	DQA[15:8]	
dfi0_rddata_en_P{0..3}[1]	BP_B1_D[8]	DQSA_C[1]	

DFI Signal	RTL Bump Name	LPDDR4X (CH A)	LPDDR4X (CH B)
dfi0_rddata_en_P{0..3}[1]	BP_B1_D[9]	DPARA[1] or DQSA_T[1]	
	BP_B1_D[10]	Unused	
	BP_B1_D[11]	Unused	
dfi0_wrdata_mask_P{0..3}[3,1]/ dfi0_rddata_dbi_W{0..3}[3,1],	BP_B1_D[12]	DM/DBIA[1]	

#### 4.1.6 LPDDR4X DFI-to-SDRAM Pin Map DWC\_LPDDR5XPHY\_NUM\_CHANNEL\_2

Table 4-4 LPDDR4X DFI-to-SDRAM Pin Map (Dual Channel, Dual Rank)

DFI Signal	RTL Bump Name	LPDDR4X (CH A)	LPDDR4X (CH B)
dfi0_address_P{0..3}[0]	BP_A[0]	CAA[0]	
dfi0_address_P{0..3}[1]	BP_A[1]	CAA[1]	
dfi0_address_P{0..3}[2]	BP_A[2]	CAA[2]	
dfi0_address_P{0..3}[3]	BP_A[3]	CAA[3]	
dfi0_address_P{0..3}[4]	BP_A[6]	CAA[4]	
dfi0_address_P{0..3}[5]	BP_A[7]	CAA[5]	
dfi0_cke_P{0..3}[0]	BP_A[4]	CKEA[0]	
dfi0_cke_P{0..3}[1]	BP_A[5]	CKEA[1]	
dfi0_cs_P{0..3}[0]	BP_A[8]	CSA[0]	
dfi0_cs_P{0..3}[1]	BP_A[9]	CSA[1]	
dfi0_dram_clk_disable_P{0..3}[0]	BP_CK0_T	CLKA_T	
dfi0_dram_clk_disable_P{0..3}[0]	BP_CK0_C	CLKA_C	
dfi1_address_P{0..3}[0]	BP_A[10]		CAB[0]
dfi1_address_P{0..3}[1]	BP_A[11]		CAB[1]
dfi1_address_P{0..3}[2]	BP_A[12]		CAB[2]
dfi1_address_P{0..3}[3]	BP_A[13]		CAB[3]
dfi1_address_P{0..3}[4]	BP_A[16]		CAB[4]
dfi1_address_P{0..3}[5]	BP_A[17]		CAB[5]
dfi1_cke_P{0..3}[0]	BP_A[14]		CKEB[0]
dfi1_cke_P{0..3}[1]	BP_A[15]		CKEB[1]

DFI Signal	RTL Bump Name	LPDDR4X (CH A)	LPDDR4X (CH B)
dfi1_cs_P{0..3}[0]	BP_A[18]		CSB[0]
dfi1_cs_P{0..3}[1]	BP_A[19]		CSB[1]
dfi1_dram_clk_disable_P{0..3}[0]	BP_CK1_T		CLKB_T
dfi1_dram_clk_disable_P{0..3}[0]	BP_CK1_C		CLKB_C
dfi_reset_n	BP_MEMRESET_L	RESET_N	
dfi0_wrdata_P{0..3}/ dfi0_rddata_W{0..3}[{23:16,7:0}], dfi0_wrdata_cs_P{0..3}[1:0] dfi0_rddata_cs_P{0..3}[1:0],	BP_B0_D[7:0]	DQA[7:0]	
dfi0_rddata_en_P{0..3}[0]	BP_B0_D[8]	DQSA_C[0]	
dfi0_rddata_en_P{0..3}[0]	BP_B0_D[9]	DPARA[0] or DQSA_T[0]	
	BP_B0_D[10]	Unused	
	BP_B0_D[11]	Unused	
dfi0_wrdata_mask_P{0..3}[2,0]/ dfi0_rddata_dbi_W{0..3}[2,0] ,	BP_B0_D[12]	DM/DBIA[0]	
dfi0_wrdata_P{0..3}/ dfi0_rddata_W{0..3}[31:24,15:8] , dfi0_wrdata_cs_P{0..3}[3:2], dfi0_rddata_cs_P{0..3}[3:2],	BP_B1_D[7:0]	DQA[15:8]	
dfi0_rddata_en_P{0..3}[1]	BP_B1_D[8]	DQSA_C[1]	
dfi0_rddata_en_P{0..3}[1]	BP_B1_D[9]	DPARA[1] or DQSA_T[1]	
	BP_B1_D[10]	Unused	
	BP_B1_D[11]	Unused	
dfi0_wrdata_mask_P{0..3}[1,1]/ dfi0_rddata_dbi_W{0..3}[3,1] ,	BP_B1_D[12]	DM/DBIA[1]	
dfi1_wrdata_P{0..3}/ dfi1_rddata_W{0..3}[{23:16, 7:0}], dfi1_wrdata_cs_P{0..3}[1:0], dfi1_rddata_cs_P{0..3}[1:0],	BP_B2_D[7:0]		DQB[7:0]
dfi1_rddata_en_P{0..3}[0]	BP_B2_D[8]		DQSB_C[0]

DFI Signal	RTL Bump Name	LPDDR4X (CH A)	LPDDR4X (CH B)
dfi1_rddata_en_P{0..3}[0]	BP_B2_D[9]		DPARB[0] or DQSB_T[0]
	BP_B2_D[10]		Unused
	BP_B2_D[11]		Unused
dfi1_wrdata_mask_P{0..3}[2,0]/ dfi1_rddata_dbi_W{0..3}[2,0],	BP_B2_D[12]		DM/DBIB[0]
dfi1_wrdata_P{0..3}/ dfi1_rddata_W{0..3}[{31:24, 15:8}], dfi1_wrdata_cs_P{0..3}[3:2], dfi1_rddata_cs_P{0..3}[3:2],	BP_B3_D[7:0]		DQB[15:8]
dfi1_rddata_en_P{0..3}[1]	BP_B3_D[8]		DQSB_C[1]
dfi1_rddata_en_P{0..3}[1]	BP_B3_D[9]		DPARB[1] or DQSB_T[1]
	BP_B3_D[10]		Unused
	BP_B3_D[11]		Unused
dfi1_wrdata_mask_P{0..3}[3,1]/ dfi1_rddata_dbi_W{0..3}[3,1],	BP_B3_D[12]		DM/DBIB[1]

## 4.2 DFI Pin Swizzling

The DFI to SDRAM pin mapping, “[DFI Pin List and Pin Mapping](#)” on page 114, describes the default connection between PHY BUMP and DRAM pins.

However, some time to help floor-planning or packaging these pin connections are swizzled. PHY provides software configuration to correct the command/address and data bit swizzling before being used by PHY’s internal logic.

For DRAM DQ pins:

- Bit swizzling can be supported within DQ[7:0] and DMI/DM bits of each byte.
  - PHY bump name BP\_B\*\_D[7:0] and BP\_B\*\_D[12]
- The CSR DQnLnSel[3:0] needs to be programmed based on physical swizzling where n is 0 to 8. For programming guidance, see “[Register Descriptions](#)” on page 349.
- Byte swizzling can be supported within a 16 bit DRAM channel.

For DRAM CA pins :

- CA bits can be swizzled independently within each DFI channel. Each bit has separate control csrMapCAntoDfi. CA bits cannot be swizzled across two DFI channels.
- The CSR MapCAntoDfi needs to be programmed based on physical swizzling where n is 0 to 5(LP4X) and 6 (LP5/5X). For programming guidance, see “[Register Descriptions](#)” on page 349.
- CA[ ] bits can be swizzled independently within each DFI channel. Each bit has separate control csrMapCAntoDfi. CA [ ]bits cannot be swizzled across two DFI channels.

PHY does not support swizzling on following DRAM pins:

- CS pins
  - LPDDR5/5X mode: PHY bump name BP\_A[5:4] and BP\_A[15:14]
  - LPDDR4X mode: PHY bump name BP\_A[8] and BP\_A[19:17]
- DQS pins
  - PHY bump name BP\_B\*\_D[9:8]
- WCK pins
  - LPDDR5/5X mode: PHY bump name BP\_B\*\_D[11:10]
  - LPDDR4X mode: N/A
- CK pins
  - PHY bump name BP\_CK\*\_{T,C}.
- CKE pins
  - LPDDR4X mode: PHY bump name BP\_A[5:4] and BP\_A[15:14]

## 4.3 DFI Interface Implementation Details

Different segments of the DFI Control Interface are used depending on the target DRAM protocol. The following list indicates the parts used per protocol (and the maximum bus width allowed by the PHY in the mode):

- LPDDR4X: dfi\_address (6), dfi\_cs (2), dfi\_cke(2)
- LPDDR5/5X: dfi\_address(7), dfi\_cs(2)

## 4.4 DFI Phase Bits, Chip Selects and Data Bits

There are four phase bits for each read and write enable. Each set of read and write enable corresponds to each Byte of Data. DFI read and write enable registers organized as shown in [Table 4-5](#) for 2 BYTE configuration. For 4 BYTE configuration, for example, there will be 4 sets of read and write enable.

**Table 4-5 wr\_data\_en and rd\_data\_en encoding for DWC\_LPDDR5XPHY\_NUM\_DBYTES\_PER\_CHANNEL\_2**

Phase	3	2	1	0
Byte	1	0	1	0
dfi_{wr/rd}data_en	1	0	1	0

In DFI 1:2 mode, only phases 0 and 1 are used (signals named with \_P0 and \_P1 or \_W0 and \_W1); phases 2 and 3 are not used. In DFI 1:4 mode, all 4 phases are used.

There can be 1 or 2 chip select bits for each byte of data, the wr\_data\_cs\_n and rd\_data\_cs\_n bits are organized as shown in [Table 4-6](#) for 2 BYTE Configuration.

**Table 4-6 wr\_data\_cs\_n and rd\_data\_cs\_n encoding for DWC\_LPDDR5XPHY\_BYTES\_PER\_CHANNEL\_2 and DWC\_LPDDR5XPHY\_NUM\_RANKS\_2**

Phase	3	2	1	0
Byte	1	0	1	0
dfi_{wr/rd}data_cs	3	2	1	0

In DFI 1:2 mode, only phases 0 and 1 are used (signals named with \_P0 and \_P1 or \_W0 and \_W1); phases 2 and 3 are not used. In DFI 1:4 mode, all 4 phases are used.

There are 16 data bits (data is DDR rate) for each Byte of Data, the wr\_data and rd\_data bits are organized as shown in [Table 4-7](#) for 2 BYTE Configuration. The B0 to B3, each indicate 8 bits of data. B0 (dfi\_wrdata[7:0]) is transmitted on rising edge (even beat) and B2 (dfi\_wrdata[23:17]) is transmitted on falling edge (odd beat) for byte lane0. Same rule applies for all byte lanes.

**Table 4-7 wr\_data and rd\_data encoding for DWC\_LPDDR5XPHY\_BYTES\_PER\_CHANNEL\_2**

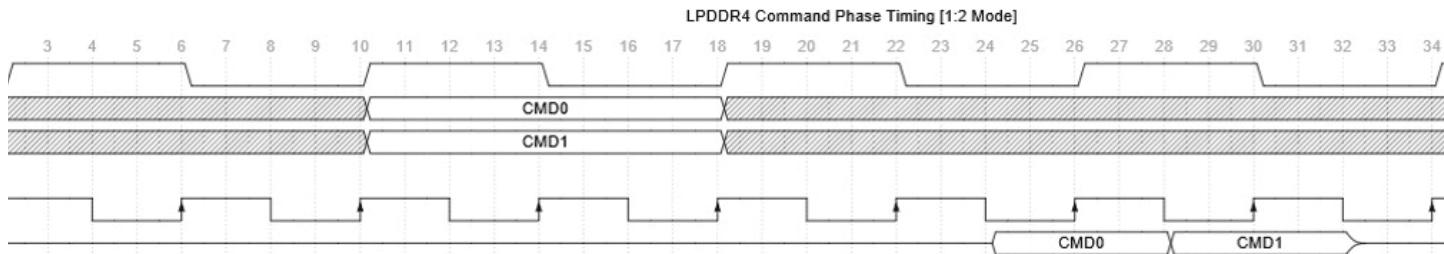
Phase	3				2				1				0			
Byte	1	0	1	0	1	0	1	0	1	0	1	0	1	0(F)	1	0(R)
dfi_{wr/rd}data	B3	B2	B1	B0	B3	B2	B1	B0	B3	B2	B1	B0	B3	B2	B1	B0

In DFI 1:2 mode, only phases 0 and 1 are used (signals named with \_P0 and \_P1 or \_W0 and \_W1); phases 2 and 3 are not used. In DFI 1:4 mode, all 4 phases are used.

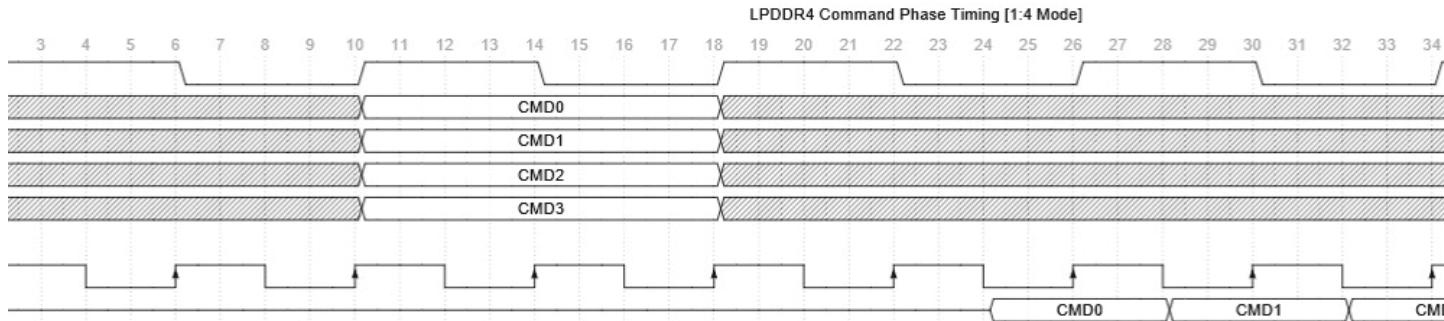
## 4.5 DFI Command Phase Timing

There can be two or four commands per DFI clock cycle, the nomenclature [CMD3:CMD0] denotes the first (even) command and the second (odd) command etc; for instance, CMD0 is driven out on the first CK cycle and CMD1 is driven on the second CK cycle and CMD2/CMD3 on the third/fourth CK cycle respectively.

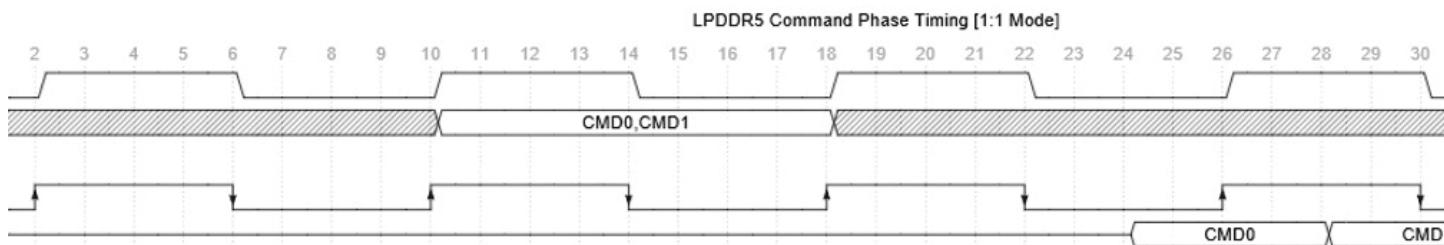
**Figure 4-1 Command Phase Timing [LPDDR4X DFI 1:2 mode]**



**Figure 4-2 Command Phase Timing [LPDDR4X DFI 1:4 mode]**



**Figure 4-3 Command Phase Timing [LPDDR5/5X 1:1 Mode]**





# 5

## Parameter Description

---

This chapter contains the following sections:

- “[PUB Parameters](#)” on page [128](#)

## 5.1 PUB Parameters

The following tables show the programmable parameters for the PUB that enable functionality. For clarity, the parameter names for parameterized signals are shown abbreviated in the table and are fully described, including the full parameter name, in the notes section at the end of the table.

### 5.1.1 DFI Control Timing Parameters

**Table 5-1 DFI Control Timing Parameters**

Parameter	Units	1:2 Mode	1:4 Mode	Description
$t_{ctrl\_delay}$ (LPDDR5/5X)	MEMCLKS	$[(PclkPtrinitVal[4:0] + 1)/8/PllMode^5] + 1.5 + AcInPipe^1 + MISC^3 + Trained_AcxTxDly$	$[(PclkPtrinitVal[4:0] + 1)/8] + 1.25 + (AcInPipe^1 + MISC^3 + Trained_AcxTxDly)$	Specifies the number of MEMCLKs after a change in the DFI control signals (as sampled by DfiClk in the PHY) that the change is reflected on the PHY-DRAM interface
$t_{ctrl\_delay}$ (LPDDR4X)	MEMCLKS	$[(PclkPtrinitVal[4:0]+1)/4/PllMode^6] + 3 + (AcInPipe^{1*2}) + (MISC^{3*2}) + Trained_AcxTxDly$	$[(PclkPtrinitVal[4:0] + 1)/2/PllMode^6] + 5 + (AcInPipe^{1*4}) + (MISC^{3*4}) + Trained_AcxTxDly$	Specifies the number of MEMCLKs after a change in the DFI control signals (as sampled by DfiClk in the PHY) that the change is reflected on the PHY-DRAM interface
$t_{cmd\_lat}$	MEMCLKS	0	0	Specifies the number of DFI clocks after the dfi_cs signal is asserted until the associated CA signals are driven.

Table Notes:

1. AcInPipe= csrAcInPipeEn
2. DFI timing value is increased as per section “[Support LPDDR5 Low Frequency](#)” on page [222](#)
3. MISC is the setting of DWC\_LPDDR5XPHY\_PIPE\_DFI\_MISC
4. For changes to \_P{1-3} DFI control signals, add 1-3 additional MEMCLK respectively.
5. LP5/5x: PllMode=2 if PlkBypass=1. Otherwise, PllMode=1
6. LP4x: PllMode=1 If PLL is in X4 Mode. Otherwise, PllMode=2
7. For  $t_{ctrl\_delay}$  (LPDDR5/5X), when AC 1/4 clock rate is enabled, 0.85 MEMCLKs of delay will be added.

## 5.1.2 DFI Write Parameters and Write Timing Parameters

### 5.1.2.1 DFI Write Parameters

**Table 5-2 DFI Write Data Programmable Parameters**

Parameter	Description
phydbi_mode	Determines which device generates DBI and inverts the data. <ul style="list-style-type: none"> <li>■ ‘b0 = DBI generation and data inversion performed in the MC.</li> <li>■ ‘b1 = Not supported.</li> </ul>

### 5.1.3 DFI Write Timing Parameters

**Table 5-3 DFI Write Timing Parameters**

Parameter <sup>2</sup>	LPDDR4X			LPDDR5			Description
	Units	1:2 Mode	1:4 mode	Units	1:2 Mode	1:4 Mode	
tphy_wrcgap (LPDDR4X Only)	MEMCLKs	5	5	NA	NA	NA	This parameter specifies the minimum number of additional MEMCLK that are required between commands when changing the target chip select that is driven on the dfi_wrdata_cs signal. In LPDDR5, tphy_wckcsgap should be used, instead.
tphy_wrcslat	MEMCLKs	$WL^4 - 5 + (AclnPipe - DxlnPipe)^*2$	$WL^4 - 5 + (AclnPipe - DxlnPipe)^*4$	WCK	$WL^*2 - 5 + (AclnPipe - DxlnPipe)^*2$	$WL^*4 - 5 + (AclnPipe - DxlnPipe)^*2$	This parameter specifies the number of MEMCLK/WCK from the time that a write command is sent on the DFI control interface and when the associated dfi_wrdata_cs signal is asserted.
tphy_wrdata	MEMCLKs	2	2	WCK	2	2	This parameter specifies the number of MEMCLK/WCK from the time that the dfi_wrdata_en signal is asserted and when the associated write data is driven on the dfi_wrdata signal. The parameter adjusts the relative time between enable and data transfer with no effect on performance.

<b>Parameter<sup>2</sup></b>	LPDDR4X			LPDDR5			<b>Description</b>
	<b>Units</b>	<b>1:2 Mode</b>	<b>1:4 mode</b>	<b>Units</b>	<b>1:2 Mode</b>	<b>1:4 Mode</b>	
tphy_wrlat	MEMCLKs	WL <sup>4</sup> -5 + (AcInPipe – DxInPipe) *2	WL <sup>4</sup> -5 + (AcInPipe – DxInPipe) *4	WCK	WL *2 – 5 + (AcInPipe – DxInPipe) *2	WL *4 – 5 + (AcInPipe – DxInPipe) *2	This parameter specifies the number of MEMCLK/WCK from the time that a write command is sent on the DFI control interface and when the dfi_wrdata_en signal is asserted.
tphy_wrdata_delay	MEMCLKs	tctrl_delay + 8+ BL/2 + Trained_TxDqsDly + DxInPipe*2	tctrl_delay + 8+ BL/2 + Trained_TxDqsDly + DxInPipe*4	N/A	N/A	N/A	This parameter specifies the time from dfi_wrdata_en is asserted to when the write data transfer completes on the DRAM bus.

## Table Notes:

1. For additional factors affecting Tphy\_wrcsgap, refer to section “[Rank-to-Rank Spacing](#)” on page 306.
2. DxInPipe = csrDxInPipeEn
3. AcInPipe =csrAcInPipeEn
4. WL = DRAM\_WL + tDQSS  
DRAM\_WL: DRAM CAS write latency
5. DFI timing value in increased as per section “[Support LPDDR5 Low Frequency](#)” on page 222.
6. tphy\_wrcsgap calculation is based on 0.5 tck write postamble. When a 1.5 tck write postamble is enabled, transaction spacing must increase by 1tck

## 5.1.4 DFI WCK Control Timing Parameters (LPDDR5/5X)

**Table 5-4 WCK Control Timing Parameter**

Parameter	Units	1:2 Mode	1:4 Mode	Description
twck_dis <sup>1</sup>	tWCK	WCK always on enabled case: 1) roundup(tWCKSTOP/tWCK) - 3 OR 2) roundup(tCSLCK/tWCK) - 3 WCK always on disabled case: 3) roundup(tWCKPST/tWCK) - 4	WCK always on enabled case: 1) roundup(tWCKSTOP/tWCK) - 3 OR 2) roundup(tCSLCK/tWCK) - 3 WCK always on disabled case: 3) roundup(tWCKPST/tWCK) - 4	Defines the number of clock cycles between the last command (1) for CAS(WS_OFF) and 2) for PDE/SR-PD/DSM) without a WCK synchronization required (assuming no command issued) or any command that disables the WCK to when the dfi_wck_en signal is disabled.
tphy_wckcsgap <sup>6</sup>	tWCK	3 * 2	1 * 4	This parameter specifies the minimum number of additional idle time that are required between CAS_WS_RD/CAS_WS_WR commands when changing the target chip select that is driven on the dfi_wck_cs signal.
twck_en_fs <sup>1</sup>	tWCK	tWCKENL_FS *2 - 4 + (AclnPipe - DxlnPipe)	tWCKENL_FS *4 - 4 + (csrAclnPipeEn - csrDxlnPipeEn)*2	Defines the number of clocks between the CAS_WS_FS command to when the dfi_wck_en signal is driven.
twck_en_rd <sup>1</sup>	tWCK	tWCKENL_RD *2 - 4 + (AclnPipe - DxlnPipe)	tWCKENL_RD *4 - 4 + (AclnPipe - DxlnPipe)*2	Defines the number of clocks between the CAS_WS_RD command to when the dfi_wck_en signal is driven.
twck_en_wr	tWCK	tWCKENL_WR *2 - 4 + (AclnPipe - DxlnPipe)	tWCKENL_WR *4 - 4 + (AclnPipe - DxlnPipe)*2	Defines the number of clocks between the CAS_WS_WR command to when the dfi_wck_en signal is driven.

Parameter	Units	1:2 Mode	1:4 Mode	Description
twck_en_wrX	tWCK	0	0	Defines the number of clocks between the CAS_WS_WRX command to when the dfi_wck_en signal is driven.
twck_fast_toggle	tWCK	N/A	4	Defines the number of clock cycles between the dfi_wck_toggle being driven to TOGGLE to when the dfi_wck_toggle is driven to FAST_TOGGLE. This timing is only applicable when the WCK transitions from the slow to fast toggle i.e. in WCK:CK=4:1 mode. Otherwise, this timing parameter should be set to 0x0.
twck_toggle	tWCK	(tWCKPRE_Static*2)	(tWCKPRE_Static*4)	Defines the number of clock cycles between the dfi_wck_en signal being enabled to when the dfi_wck_toggle signal is driven to TOGGLE.
twck_toggle_cs	tWCK	0	0	Defines the number of clock cycles between a read or write command to when the dfi_wck_cs signal must be stable. This timing is applicable when the WCK is synchronized for multiple CS's and commands are to different CS's. During WCK synchronization, the CS should be static from the CAS command to the completion of the synchronization sequence.

Parameter	Units	1:2 Mode	1:4 Mode	Description
twck_toggle_post	tWCK	(BL/n_max - BL/n_min ) *2 + roundup(tWCKPST)	(BL/n_max - BL/n_min) *4 + roundup(tWCKPST)	Defines the number of clock cycles after a read or write command data burst completion during which the WCK must remain in the current toggle state. During this time, the dfi_wck_cs signal must also remain stable.
Twck_delay	tWCK	Roundup(tctrl_delay*2+8+ max across all bytes(csrTxWckDly[10:6]/2)	Roundup(tctrl_delay*4+8+ max across all bytes(csrTxWckDly[10:6]/2)	This parameter specifies the maximum_delay from dfi_wck_en de-assertion to when WCK transfer completes on the DRAM bus. For more detail, see <a href="#">"Entering DFI LP State" on page 317</a> .

Table Notes:

1. DFI timing value increased as per section ["Support LPDDR5 Low Frequency" on page 222](#).
2. DxInPipe refers to the number of bits that are '1' in DxInPipe = csrDxInPipeEn
3. AcInPipe refers to the number of bits that are '1' in AcInPipe = csrAcInPipeEn
4. tWCKENL\_{RD,WR,FS}, tWCKPRE\_Static, tWCKPST are LPDDR5 JEDEC timing parameters.
5. WCK = nominal WCK period (tWCK)
6. For additional factors affecting tphy\_wckcsgap, refer to ["Rank-to-Rank Spacing" on page 306](#).

### 5.1.5 DFI Read Parameters

Table 5-5 DFI Read Data Programmable Parameters

Parameter	Description
phydbi_mode	Determines which device generates DBI and inverts the data. <ul style="list-style-type: none"> <li>■ 'b0 = DBI generation and data inversion performed in the MC.</li> <li>■ 'b1 = Not supported</li> </ul>

## 5.1.6 DFI Read Timing Parameters

**Table 5-6 DFI Read Data Timing Parameters**

Parameter	LPDDR4X			LPDDR5			Description
	Units	1:2 Mode	1:4 Mode	Units	1:2 Mode	1:4 Mode	
tphy_rd_csgap	MEMCLKs	6	6	WCK	N/A	N/A	Specifies the minimum number of additional MEMCLK/WCK required between commands when changing the target chip select driven on the dfi_rddata_cs signal. This parameter needs to be supported in the MC transaction-to-transaction timing. In LPDDR5, tphy_wckcsgap is used instead.
tphy_rd_cslat	MEMCLKs	$RL^2 - 5 + (AcInPipe - DxInPipe) * 2$	$RL^2 - 5 + (AcInPipe - DxInPipe) * 4$	WCK	$RL^2 * 2-5 + (AcInPipe - DxInPipe) * 2$	$RL^2 * 4-5 + (AcInPipe - DxInPipe) * 2$ Data rate > 3200 Mbps:: $RL^2 * 4-13 + (AcInPipe - DxInPipe) * 2$	Specifies the number of MEMCLKs/WCK between when a read command is sent on the DFI control interface and when the associated dfi_rddata_cs signal is asserted.
tphy_rdl_at	MEMCLKs	Refer to “PHY Latency” on page 304 trained	Refer to “PHY Latency” on page 304 trained	WCK	Refer to “PHY Latency” on page 304	Refer to “PHY Latency” on page 304	Specifies the maximum number of MEMCLK/WCK allowed from the assertion of the dfi_rddata_en signal to the assertion of each of the corresponding bits of the dfi_rddata_valid signal.

Parameter	LPDDR4X			LPDDR5			Description
	Units	1:2 Mode	1:4 Mode	Units	1:2 Mode	1:4 Mode	
trddata_en	MEMCLKs	$RL^2 - 5 + (AcInPipe - DxInPipe)^*2$	$RL^2 - 5 + (AcInPipe - DxInPipe)^*4$	WCK	$RL^2*2-5 + (AcInPipe - DxInPipe)^*2$	Data rate <= 3200 Mbps:: $RL^2 * 4-5 + (AcInPipe - DxInPipe)^*2$	Specifies the number of MEMCLK/WCK from the assertion of a read command on the DFI to the assertion of the dfi_rddata_en signal.

Table Notes:

1. tphy\_rdcsgap calculation is based on 2 tck read preamble. tphy\_rdcsgap calculation is based on 1.5 tck read postamble.
2. RL = DRAM\_RL Where:
  - DRAM\_RL: DRAM CAS read latency
3. DxInPipe refers to the number of bits that are '1' in DxInPipe = csrDxInPipeEn
4. AcInPipe refers to the number of bits that are '1' in AcInPipe = csrAcInPipeEn
5. DFI timing value is increased as per section "[Support LPDDR5 Low Frequency](#)" on page [222](#).
6. For additional factors affecting tphy\_rdcsgap, refer to "[Rank-to-Rank Spacing](#)" on page [306](#).

### 5.1.7 DFI Update Timing Parameters

Table 5-7 DFI Update Timing Parameters

Parameter	Units	1:2 Mode	1:4 Mode	Description
tphyupd_type0	DfiClk	$25 + MISC^2 + csrPhyUpdReqDelay$	$25 + MISC^2 + csrPhyUpdReqDelay$	Specifies the maximum number of DfiClk cycles that the dfi_phyupd_req signal may remain asserted after the assertion of the dfi_phyupd_ack signal for dfi_phyupd_type = 0x0. The dfi_phyupd_req signal may deassert at any cycle after the assertion of the dfi_phyupd_ack signal.

Parameter	Units	1:2 Mode	1:4 Mode	Description
$t_{phyupd\_resp}$ (max)	DfiClk	508 - MISC*2	508 - MISC*2	Specifies the maximum number of DfiClk cycles after the assertion of the dfi_phyupd_req signal to the assertion of the dfi_phyupd_ack signal.
$t_{ctrlupd\_min}$	DfiClk	$2 + \text{MISC}^*2 + \text{csrCtrlUpdReqDelay}$	$2 + \text{MISC}^*2 + \text{csrCtrlUpdReqDelay}$	Specifies the minimum time for dfi_ctrlupd_req assertion
$t_{ctrlupd\_max}$	DfiClk	$25 + \text{MISC}^*2 + \text{csrCtrlUpdAckDelay} + \text{csrCtrlUpdReqDelay}$	$25 + \text{MISC}^*2 + \text{csrCtrlUpdAckDelay} + \text{csrCtrlUpdReqDelay}$	Specifies the maximum time for dfi_ctrlupd_req assertion

Table Notes:

- $t_{phyupd\_type0}$  value shown in table is with csrPhyUpdAckDelay = 0. This is a debug only CSR.
- The value  $t_{phyupd\_resp(max)}$  in table is with csrPhyUpdAckDelay = 0. This is a debug only CSR.
- The  $t_{phyupd\_resp}$  value in table is with default setting of csrDFIPHYPUPDRESP.
- MISC is the setting of DWC\_LPDDR5XPHY\_PIPE\_DFI\_MISC

## 5.1.8 DFI Low Power Parameters and Low Power Control Timing Parameters

### 5.1.8.1 DFI Low Power Parameters

Table 5-8 DFI Low Power Parameters

Parameter	Description
DFI_low_power_version	DFI low power version parameter for connecting DFI 5 with older version DFI. <ul style="list-style-type: none"> <li>■ 0: Connecting with DFI 5 (and above) Low Power Interface.</li> <li>■ 1: Connecting with older version to DFI 5 Low Power Interface. [Not Supported]</li> </ul>

### 5.1.8.2 DFI Low Power Control Timing Parameters

Table 5-9 DFI Low Power Control Timing Parameters

Parameter	Units	1:2 Mode	1:4 Mode	Description
$t_{lp\_resp}$	DfiClk	$2 + 2^*\text{MISC}^a$	$2 + 2^*\text{MISC}^a$	Specifies the maximum number of DfiClk cycles after the assertion of the dfi_lp_ctrl_req or dfi_lp_data_req signal to the assertion of the dfi_lp_ack signal
$t_{lp\_ctrl\_wake\_up}$	DfiClk	$4 + \text{csrLpCtrlAckDelay} + 2^*\text{MISC}^a$	$4 + \text{csrLpCtrlAckDelay} + 2^*\text{MISC}^a$	Specifies the maximum number of DfiClk cycles that the dfi_lp_ctrl_ack signal may remain asserted after the de-assertion of the dfi_lp_ctrl_req signal. The dfi_lp_ctrl_ack signal may de-assert at any cycle after the de-assertion of the dfi_lp_ctrl_req signal. Exceeding the maximum is not considered an error condition.

Parameter	Units	1:2 Mode	1:4 Mode	Description
$t_{lp\_data\_wakeup}$	DfiClk	$2 + csrLpDataAckDelay + 2*MISC^a$	$2 + csrLpDataAckDelay + 2*MISC^a$	Specifies the maximum number of DfiClk cycles that the dfi_lp_data_ack signal may remain asserted after the de-assertion of the dfi_lp_data_req signal. The dfi_lp_data_ack signal may de-assert at any cycle after the de-assertion of the dfi_lp_data_req signal. Exceeding the maximum is not considered an error condition.

- a. MISC is the setting of DWC\_LPDDR5XPHY\_PIPE\_DFI\_MISC
- b. Refer to “DFI LP Control Interface” on page 322 for different Power saving modes and “tlp\_ctrl\_wakeup” timings.

## 5.1.9 DFI Error Timing Parameters

**Table 5-10 DFI Error Timing Parameters**

Parameter	Units	Value	Description
$t_{error\_resp}$	DfiClk	$2 + MISC$	Specifies the maximum number of DfiClk cycles that may occur from the “DFI error condition” and the assertion of the dfi_error signal.

**Table Notes:**

1. Refer to “DFI Error Interface” on page 326 for details on “DFI error condition”.
2. MISC is the setting of DWC\_LPDDR5XPHY\_PIPE\_DFI\_MISC

### 5.1.10 DFI PHY MASTER Timing Parameters

**Table 5-11 DFI PHY MASTER Timing Parameters**

Parameter	Units	1:2 Mode	1:4 Mode	Description
$t_{phymstr\_resp}$ (max)	DfiClk	2042 - 2*MISC <sup>a</sup>	2042 - 2*MISC <sup>a</sup>	For the recommended setting of csrPhyMstrMaxReqToAck (0x4), specifies the maximum number of DfiClk cycles after the dfiphymstr_req signal asserts to the assertion of the dfi_phymstr_ack signal.
$t_{phymstr\_type0}$ (max)	DfiClk	$t_{init\_complete}$ (RetrainOnly) + 30	$t_{init\_complete}$ (Retrain Only) + 30	Specifies the maximum number of DfiClk cycles that the dfi_phymstr_req signal may remain asserted after the assertion of the dfi_phymstr_ack signal for dfi_phymstr_type = 0x0. The dfi_phymstr_req signal may de-assert at any cycle after the assertion of the dfi_phymstr_ack signal.
$t_{phymstr\_type\{1-3\}}$	-	-	-	Not supported

a. MISC is the setting of DWC\_LPDDR5XPHY\_PIPE\_DFI\_MISC

### 5.1.11 DFI Frequency Change Parameters

**Table 5-12 DFI Frequency Change Parameters**

Parameter	Units	1:2 Mode	1:4 Mode	Description
$t_{init\_start}$ (max)	DfiClk	Refer to Table Max $t_{init\_complete}$		Specifies maximum time from assertion of dfi_init_start to de-assertion of dfi_init_complete for a frequency change operation.
$t_{init\_complete}$ (max)	DfiClk	Refer to Table Max $t_{init\_complete}$		Specifies maximum time from de-assertion of dfi_init_start to assertion of dfi_init_complete for a frequency change operation.

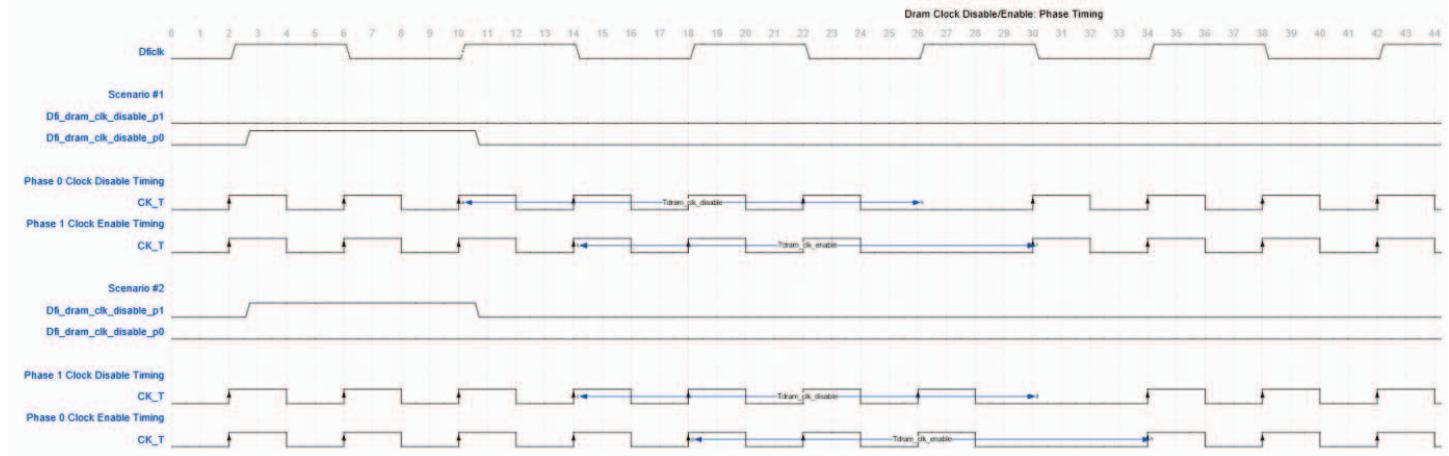
### 5.1.12 DFI DRAM Clock Disable Parameters

**Table 5-13 DFI DRAM Clock Disable Parameters**

Parameter	Unit	Value		Description
		1:2 Mode	1:4 Mode	
$t_{dram\_clk\_disable}$	MEMCLKs	tctrl_delay	tctrl_delay	Specifies the time from assertion of dfi_dram_clk_disable until clock is low at DRAM. This specification does not include board and internal DRAM delays.
$t_{dram\_clk\_enable}$	MEMCLKs	tctrl_delay	tctrl_delay	Specifies the time from deassertion of dfi_dram_clk_disable until first rising edge of clock at DRAMs. This specification does not include board and internal DRAM delays.

## Table Notes:

1. The dfi\_dram\_clk\_disable\_P\* must stay de-asserted when DFI Init / Frequency Change / PHYMSTR/ CtrlUpd/ PhyUpd are active.
2. For LPDDR5 mode, only dfi\_dram\_clk\_disable\_P0 is valid.

**Figure 5-1 Tdram\_clk\_enable/Tdram\_clk\_disable Timing Example**



# 6

## Clocks, Resets, Power Domains, Initialization, and Power Management

---

This chapter contains the following sections:

- “Clocks” on page 142
- “Resets” on page 146
- “Power Domains” on page 150
- “PHY Initialization” on page 151
- “Power Management” on page 157
- “DFI Frequency Change” on page 171b

## 6.1 Clocks

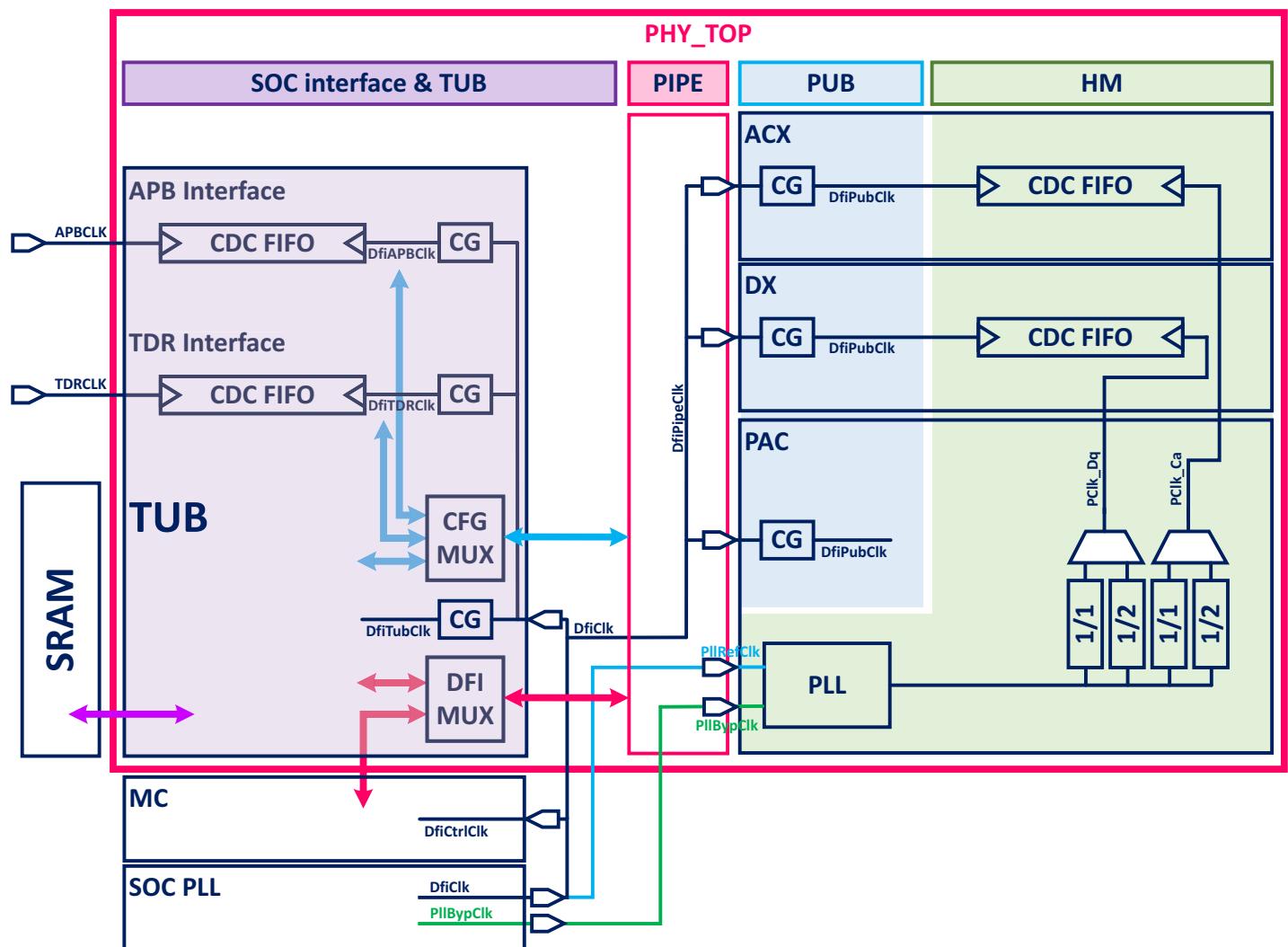
### 6.1.1 PHY Clocks

The PHY operational clocks include:

- APB Clock (APBCLK)
- DFI Clock (DfiClk)
- Phy High speed Clock (Pclk\_Ca and Pclk\_Dq)
- Phy High speed Bypass Clock (PllBypClk)
- PLL Reference Clock(PllRefClk)

Below is a high-level diagram of the PHY clock distribution.

**Figure 6-1 PHY Clock Distribution Block Diagram, Single Channel**



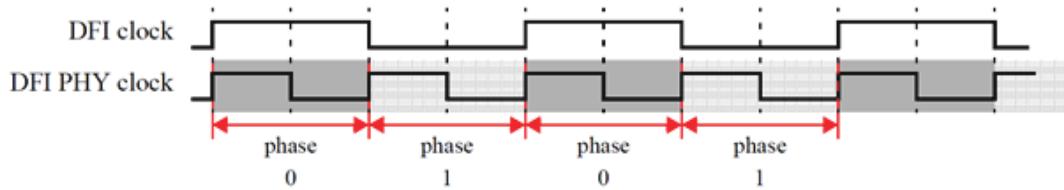
### 6.1.1.1 DfiClk

The DfiClk is generated externally and is used by the PHY as a primary mission mode clock input for the PUB logic.

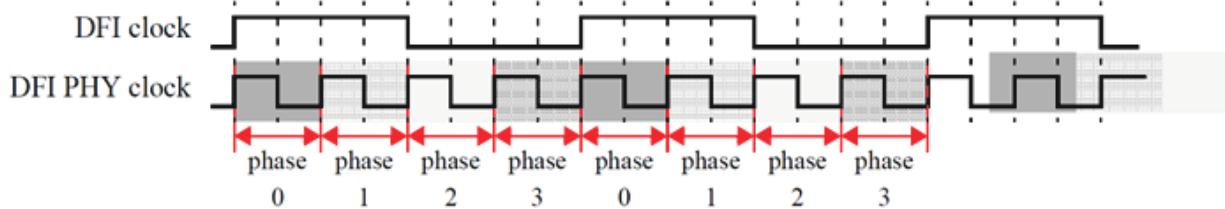
In LPDDR4X Mode, The DfiClk supplied to the PHY must be set to half or one quarter of the desired memory (MEMCLK or DFI PHY clock) frequency, which is equivalent to one quarter and one eighth the data rate, respectively.

In LPDDR5/5X Mode, The DfiClk supplied to the PHY is equal to the desired memory (MEMCLK) frequency.

**Figure 6-2 DFI\_FREQ\_RATIO=1:2**



**Figure 6-3 DFI\_FREQ\_RATIO=1:4**



The DfiClk is required to be a single-ended, full-CMOS VDD level input.

The maximum operational DfiCLK frequency = DATA\_RATE/4 in DFI 1:2 mode. It is DATA\_RATE/8 in DFI 1:4 mode.

### 6.1.1.2 DFI\_FREQ\_RATIO = 1:4 LPDDR4X Operation

In both DFI 1:2 and DFI 1:4 mode, the controller interfaces directly to the DFI interface on the PHY. There is no shim between the DFI interface and the internal PUB logic. The `dfi0_*_P0`, `dfi0_*_P1`, `dfi0_*_P2`, and `dfi0_*_P3` signals are used to provide the input information for four consecutive MEMCLK cycles, and the output from four consecutive MEMCLK cycles is provided on `dfi0_*_P0/W0`, `dfi0_*_P1/W1`, `dfi0_*_P2/W2`, and `dfi0_*_P3/W3` output signals respectively. All `dfi1_*` signals behave similarly to their `dfi0_*` namesake.

For timing diagram example, see [Figure 6-8](#) on page [155](#).

### 6.1.1.3 DFI\_FREQ\_RATIO = 1:2 LPDDR4X Operation

In both DFI 1:2 and DFI 1:4 mode, the controller interfaces directly to the DFI interface on the PHY. There is no shim between the DFI interface and the internal PUB logic. The `dfi0_*_P0` and `dfi0_*_P1` signals are used to provide the input information for two consecutive MEMCLK cycles, and the output from two

consecutive MEMCLK cycles is provided on dfi0\_\*\_P0/W0, and dfi0\_\*\_P1/W1 output signals. In 1:2 mode signals named dfi0\_\*\_P2/W2/P3/W3 on the DFI interface are unused. All dfi1\_\* signals behave similarly to their dfi0\_\* namesake.

For timing diagram example, see [Figure 6-8](#) on page [155](#).

#### 6.1.1.4 PllBypClk

The PllBypClk clock is generated externally and is used when PHY is in PLL-bypass mode. The PllBypClk is required as a single-ended, full-CMOS level input on VDD. The PllBypClk is used as the half bit-rate clock inside the PHY.

PllBypClk requirements:

Min = (Minimum supported data rate)/2

Max = 1600 MHz (for 3200 Mbps)

When the PLL-bypass feature is not needed, PllBypClk can be tied low or gated low.

#### 6.1.1.5 PllRefClk

The PllRefClk clock is generated externally and is used as reference clock by PLL oscillator when PLL is in mission mode.

The PllRefClk is required to be a single-ended, full-CMOS VDD level input.

**Table 6-1 PllRefClk**

	DFI 1:2	DFI 1:4
PllRefClk (Frequency)	DATA_RATE/4	DATA_RATE/8



The PllRefClk must always be the same frequency as DFIClk and be driven from same source.

#### 6.1.1.6 APBCLK

The APB Clock is generated externally and is synchronous to the APB configuration bus. All signals on the APB interface are synchronous to the APBCLK. Internally to the TUB, APB transactions are synchronized into DfiApbClk domain to allow APBCLK to operate at a different frequency than DfiClk.

The APBCLK frequency must be always less or equal to DfiClk frequency for each PState. For the purpose of generating and testing out timing constraints, it is tested up to 1200 MHz.

#### 6.1.2 DFx Clocks

- Test Data Register Clock (TDRCLK)
- ATPG Scan Clocks (scan\*clk)
  - scan\_DlyTestClk, scan\_clk, scan\_UcClk

### 6.1.2.1 TDRCLK

The Test Data Register clock (TDRCLK) is an externally generated DFT clock for the PHY TDR chain. In test modes, the TDR chains can be used to read and write PHY registers.

Max speed supported is 300 MHz.



**Note** The frequency(DfiClk) >= frequency(TDRCLK) in order to access registers with TDR interface

### 6.1.2.2 scan\_clk

The scan related clocks are for test modes only. Single scan\_clk port is provided for each internally generated functional clock domain in the PHY Hard Macro. For more details, refer to Implementation Guide section “ATPG Clocks and Control Logic”.

## 6.2 Resets

The PHY supports cold and warm resets. The PHY receives the following control signals for various reset states:

- BP\_PWROK
  - Reset (synchronous reset used for most logic in the PHY)
  - Reset\_Async (asynchronous reset used for ARC Processor, and some logic/circuits in hard macros)

The PHY also supports the reset of the APB interface though the following:

- PRESETn\_APB

The following reset rules must be followed:

- Reset is synchronous to DfiClk.
- Reset must have a minimum pulse width of 64 DfiClks.
- Reset and Reset\_Async must be asserted/de-asserted in a functionally equivalent manner allowing for timing-related differences.
- PRESETn\_APB is synchronous to APBCLK.
- PRESETn\_APB must have a minimum pulse width of 16 APBCLKs.
  - The asserting edge(1->0) transition of PRESETn\_APB must occur at least 16 APBCLKs before the de-asserting edge of Reset.
- After the deassertion of both Reset and PRESETn\_APB, the PHY must receive at least 64 DfiClk pulses prior to any APB transaction or an assertion of dfi\_init\_start.
- The de-asserting edge of PRESETn\_APB may occur after the de-asserting edge of Reset.
- PHY TDR Interface must be reset 1us prior to “Reset” de-assertion.
- WRSTN is synchronous to TDRCLK. As per JTAG specs, WRSTN may be asynchronously asserted, but must be synchronously de-asserted.
- APB transactions are allowed only after the PHY and the APB interface have exited reset - Reset and PRESETn\_APB have both been de-asserted.
- BP\_PWROK is asynchronous.
- During Cold Reset, the reset signals (Reset and Reset\_Async) must be asserted at least 8 DfiClks prior to the assertion of BP\_PWROK.
- During Cold Reset, the reset signals (Reset and Reset\_Async) must remain asserted for at least 64 DfiClks after the assertion of BP\_PWROK.
- After the de-assertion of BP\_PWROK, the DfiClk must cycle for at least 16 DfiClks.

### 6.2.1 Warm Reset

Warm reset occurs when the Reset signal becomes asserted while the BP\_PWROK signal is already asserted. On a warm reset, the PHY output is not deterministic.



**Note** Warm Reset, started by asserting the Reset input pin while the BP\_PWROK pin is asserted, resets all the digital logic in the PHY (including the registers). Cold Reset is the same as Warm Reset with the exception that all the I/O bumps are also immediately forced to their inactive state.

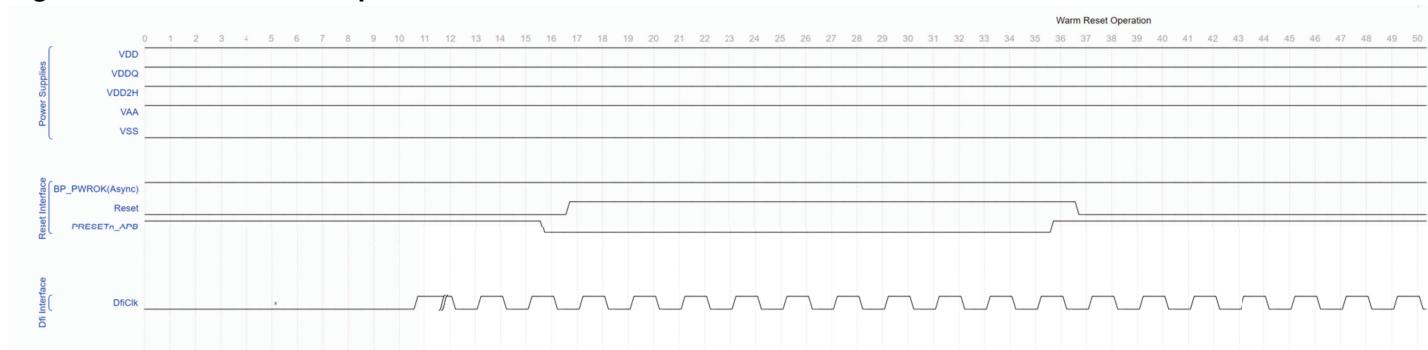
On reset assertion, the PHY stops, asserts internal inits and stops its internal clocks. If the PHY is sequenced to LP3 and BP\_PWROK==0, which enables Retention, the PHY ensures that the CS and MEMRESET signals are protected from the warm reset signal going x (unknown) or asserting.

Otherwise when the PHY and DRAM are in warm reset, the bus states are not guaranteed. While Reset is asserted, the APB and DFI interfaces are off line.

Warm reset also resets the uCtrl and the SRAM content is not guaranteed. After Reset de-asserts, the PHY must be fully reinitialized before it can be placed in mission mode. After the de-assertion of Reset, the PHY must receive at least 64 DFICLK pulses prior to any APB transaction or the assertion of dfi\_init\_start.

Figure 6-4 shows the Warm Reset operation.

**Figure 6-4 Warm Reset Operation**



## 6.2.2 Cold Reset

A cold reset occurs when, at power-up, the BP\_PWROK signal is low and Reset signal is high.

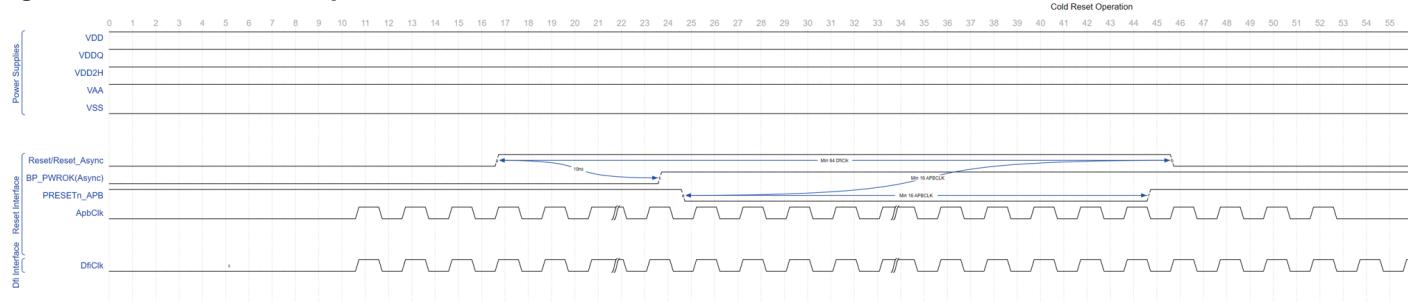


**Note** Cold Reset resets all the digital logic in the PHY, including the registers, and immediately forces all the I/O bumps to their inactive state.

Once the BP\_PWROK is asserted (and Reset is still asserted), DfiClk synchronously switches to any legal input frequency.

After power-up, subsequent cold resets can be signaled to the PHY by de-asserting BP\_PWROK and asserting Reset, with or without removing power.

Figure 6-5 shows Cold Reset operation.

**Figure 6-5 Cold Reset Operation**

The PHY receives the following control signals for various reset states:

### 6.2.3 Reset

1. Resets the logic (on DfiClk) synchronously in PUB and Hard Macro.
2. Reset must have a minimum pulse width of 64 valid DfiClks.
3. Reset is asynchronously asserted but synchronously (DfiClk ) de-asserted.

Reset input signal can be asserted at any time. When Reset input is asserted, PHY outputs behave as below:

- During power-up (ColdReset): MEMRESET\_L and CS outputs are forced 0 and other outputs are at their power-on defaults.
- During retention Mode: MEMRESET\_L output is forced 1, CS outputs are forced 0 and other outputs are un-driven, for example, not deterministic.
- During Normal Mode (WarmReset): MEMRESET\_L output is forced 1, CS outputs are forced 0 and other PHY outputs are un-driven, for example, not deterministic.

#### 6.2.3.1 On Reset assertion

- BUMP outputs of the PHY are not guaranteed
- The PHY stops its internal clocks
- Resets the PMU and the SRAM content is not guaranteed
- APB and DFI interfaces are off line

#### 6.2.3.2 After Reset de-asserts

- The PHY must be fully reinitialized before it can be placed in mission mode
- The PHY must receive at least 64 DfiClk pulses prior to any APB transaction or the assertion of dfi\_init\_start

### 6.2.4 Reset \_async

1. Reset\_async is used to control asynchronous reset pins of flops in Hard Macro and Soft IP.
2. Reset\_async is asynchronously asserted

3. In scan mode (scan\_mode=1), Reset\_async and Reset both inputs should be controlled independently to achieve better coverage.
4. In mission mode (scan\_mode=0), Reset\_async input should follow assertion and de-assertion timing of the Reset input.

## 6.2.5 PRESETn\_APB

1. Resets the APB interface logic.
2. PRESETn\_APB must have a minimum pulse width of 16 APBCLKs
3. PRESETn\_APB is synchronous to APBCLK
4. The asserting edge(1->0) transition of PRESETn\_APB must occur at least 16 APBCLKs before the deasserting edge of Reset
5. The deasserting edge of PRESETn\_APB may occur after the deasserting edge of Reset
6. APB transactions are allowed only after the PHY and the APB interface have exited reset - Reset and PRESETn\_APB have both been deasserted.

## 6.2.6 WRSTN

1. Resets the TDR/JTAG interface logic.
2. WRSTN is synchronous to TDRCLK. As per JTAG specs, WRSTN may be asynchronously asserted, but must be synchronously deasserted.
3. The outputs of all TDRs in the PHY must be fully reset (by the asynchronous signal WRSTN) for at-least 1 us before de-asserting the "Reset" pin.

## 6.3 Power Domains

**Figure 6-6 Power Supplies - LPDDR5X/5/4X PHY**

### DWC LPDDR5X/5/4X PHY Power Supplies

VDD <sup>1</sup>	VDD2H <sup>1</sup>	VAA <sup>1,2</sup>	VDDQ <sup>1</sup>
<ul style="list-style-type: none"> <li>“Core” power rail</li> <li>0.75V (nominal)</li> </ul>	<ul style="list-style-type: none"> <li>Power supply for CMOS driver, POR, and miscellaneous circuits</li> <li>LPDDR4X = 1.1V</li> <li>LPDDR5X/5 = 1.05V (same as DRAM VDD2H)</li> </ul>	<ul style="list-style-type: none"> <li>Power supply for PLL</li> <li>Options <ul style="list-style-type: none"> <li>Independent Supply 1.05V– 1.1V (nominal)</li> <li>Connect to VDD2H Supply</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>LPDDR5X/5 = 0.5V / 0.3V</li> <li>LPDDR4X = 0.6V</li> </ul>

Notes:

- Rails may power- on in any order
- The PLL supply (VAA\_VDD2H) MUST be separated from all other supplies at the die level. VAA\_VDD2H and VDD2H can be driven from the same regulator however, there must be package or PCB filtering of the VAA\_VDD2H supply

#### 6.3.1 BP\_PWROK

BP\_PWROK is a PHY input from always on VDD2H power domain. PHY requires that this input be a “clean” indication that clocks and power supplies are stable and within their specifications. The “clean” implies that signal will remain low, without glitches, from the time that power supplies are turned on until they come within specification.

This can be driven from a PAD externally or internally in SOC from VDD2H domain.

De-asserting BP\_PWRK input, MEMRESET\_L, LP5 CS and LP4X CKE pins are forced into known state based on ForceData/Enable inputs of corresponding IO.

When IO retention feature is not required, BP\_PWROK input can be connected to VIO\_TIEHI output port for example, tied high permanently.

#### 6.3.2 Controlled Power off Considerations

When power supplies are turned off, the following requirements are required for the system.

- For non IO retention usage with BP\_PWROK =1 all the time
  - Before powering off the SDRAM or PHY power rails, SOC should assert PHY Reset Input
  - SOC must follow "SDRAM's Power Off Sequence" defined in JEDEC specification.
- For IO retention usage with BP\_PWROK controllable by the system
  - Before powering off the SDRAM or PHY power rails, SOC should assert PHY Reset Input
  - Before powering off the SDRAM or PHY power rails, SOC should de-asserted BP\_PWROK

- SOC must follow "SDRAM's Power Off Sequence" defined in JEDEC specification.

## 6.4 PHY Initialization

This document only covers high level overview of initialization process. As part of the release directory structure, source code in C is provided that shows exactly what steps need to be executed during PHY initialization depending on PHY configuration. The code is in the phyinit directory in the following location:

```
<PHY Install Directory>/synopsys/<technology>_<process>/Latest/phyinit/Latest
```

By default, the provided code prints all the register accesses (complete with register address and data) necessary for all parts of the initialization process. Certain parts of the code need to be customized and written by user to generate a valid sequence. Complete source code documentation is provided in the following folder in the release package in form of .chm windows help file.

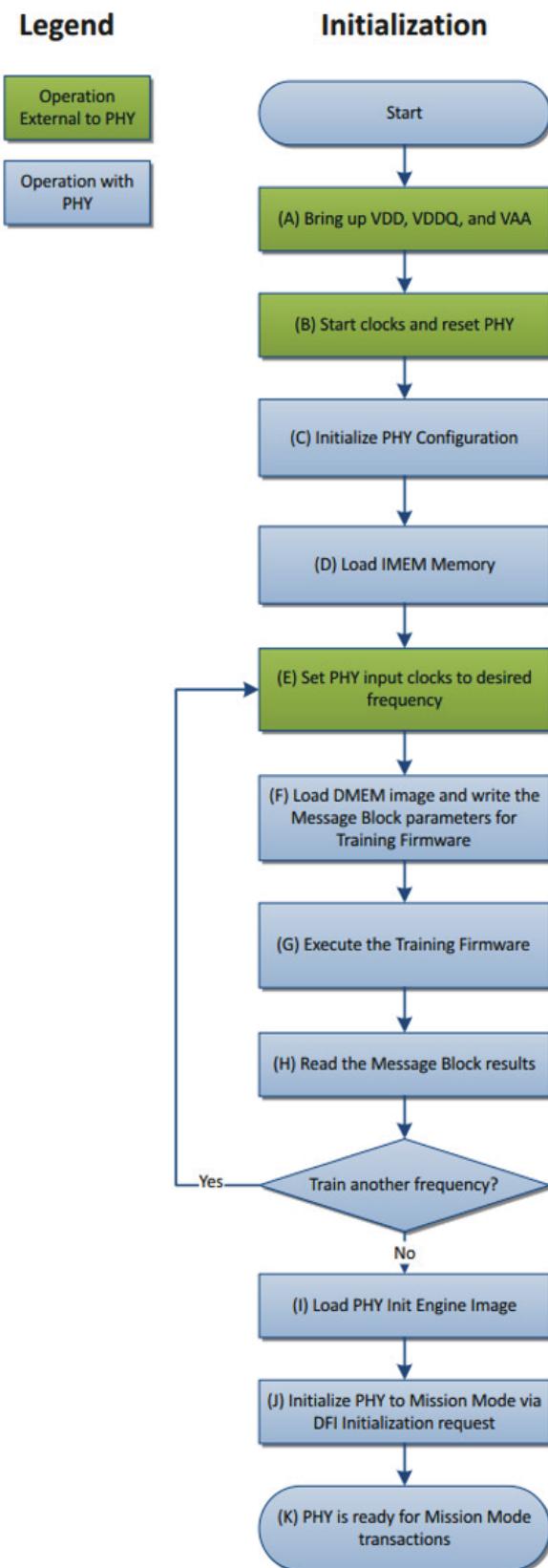
```
<PHY Install  
Directory>/synopsys/<technology>_<process>/Latest/phyinit/Latest/<protocol>/doc.
```



A quick start guide and a README is provided in phyinit folder in order to speed up initial integration. Final sequence can be obtain only when all PhyInit configuration is fully considered. This document only covers high level overview of initialization process

### 6.4.1 PHY Initialization Sequence

The following diagram shows the typical initialization process, depending on PhyInit configuration slight variations and optimization are required. Refer to PhyInit Execution flow diagram in PhyInit HTML documentation for a more detailed diagram of the steps.

**Figure 6-7 Simplified PHY Initialization Sequence**

Each step is briefly described in the next section. PhyInit documentation must be consulted for further details.

The sequence is described in detail in the following sections.

#### 6.4.1.1 (A) Bring up VDD, VDDQ, VDD2H and VAA

The first step for initializing the PHY is to apply power. The power supplies can come up and stabilize in any order. While the power supplies are coming up, all outputs will be unknown and the values of the inputs are don't cares. Once the power supplies are stable, there are rules on how the power supplies behave. See “[Power Management](#)” on page [157](#) for more information on the power supplies. For more information, refer to the `dwc_ddrphy_phyinit_A_bringupPower()` function in PhyInit documentation.

#### 6.4.1.2 (B) Start Clocks and Reset the PHY

The second step is to start the PHY clocks and apply the synchronous reset. For more information, refer to the `dwc_ddrphy_phyinit_B_startClockResetPhy()` program in the provided code in the phyinit directory.

#### 6.4.1.3 (C) Initialize PHY Configuration

The PHY supports multiple memory standards and configurations. There are configuration registers in the PHY that must be set to configure it for the correct memory type and configuration. The routine for this step details the list of configuration registers to set and how to calculate the correct values based on the system. For more information, refer to the `dwc_ddrphy_phyinit_C_initPhyConfig()` function in PhyInit documentation.

#### 6.4.1.4 (D) Load the IMEM Memory

Before the PHY training firmware can be run, the firmware program image must be loaded into the instruction memory SRAM. For more information, refer to the `dwc_ddrphy_phyinit_D_loadIMEM()` function in PhyInit documentation

#### 6.4.1.5 (E) Set the PHY input clocks to the desired frequency

Set the PHY input clocks to the desired operating frequency. Before proceeding to the next step, the clock should be stable at the new frequency. For more information on clocking requirements, see “[Clocks](#)” on page [142](#). For more information, refer to the `dwc_ddrphy_phyinit_E_setDfiClk()` function in PhyInit documentation

#### 6.4.1.6 (F) Write the Message Block parameters for the training firmware

For the firmware to run, it must be given some information about the system, the memory configuration, and the memory training steps that it is to run. There are several locations in the data memory that must be written for the training firmware to have these parameters. For more information, refer to the `dwc_ddrphy_phyinit_F_loadDMEM()` program in the provided example code for more details.

#### 6.4.1.7 (G) Execute the Training Firmware

Step G executes device initialization plus training firmware. Device initialization firmware (DevInit) performs the basic initialization of the PHY and DRAM that is required before any training stage is run. Initialization includes: resetting the DRAMs; writing the appropriate MRs; PLL locking; first ZQ calibration; first LCDL calibration; calculation of parameters to accelerate future lock operations.

There is a protocol for starting the firmware, and interacting with it to determine the training status. For more information, refer to the `dwc_ddrphy_phyinit_G_execFW()` program in the provided code in the `phyinit` directory.

#### 6.4.1.8 (H) Read the Message Block results

Once the training firmware has completed, it returns the results of the training in the data memory in the message block structure. The message block can be read to obtain these results. For more information, refer to the `dwc_ddrphy_phyinit_H_readMsgBlock()` program in the provided code in the `phyinit` directory.

#### 6.4.1.9 (I) Load PHY Init Engine Image

Load the PHY Initialization Engine registers with the provided initialization sequence. For more information, refer to the `dwc_ddrphy_phyinit_I_loadPIEImage()` program in the provided code in the `phyinit` directory.

#### 6.4.1.10 (J) Initialize the PHY to Mission Mode through DFI Initialization

Initialize the PHY to mission mode by performing a DFI initialization sequence per the DFI specification.



**Note** At the end of initial training, DRAM is in Self-Refresh power down and PHY is in LP3 state, where training firmware writes MR28 ZQ stop = 1 in LPDDR5 mode to save power when background calibration mode is selected (MR28 ZQ Mode = 0). ZQ Stop = 0 is kept when command-based calibration mode is selected (MR28 ZQ Mode = 1).

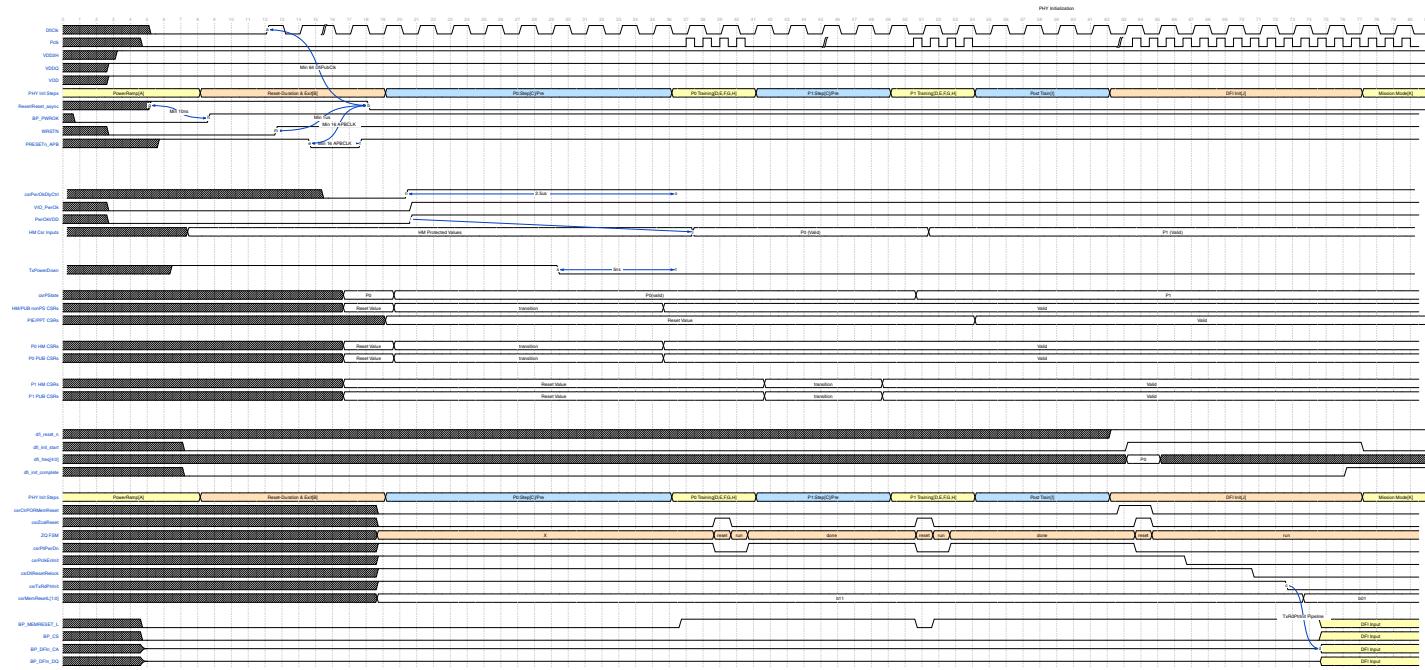
#### 6.4.1.11 (K) PHY is ready for Mission Mode transactions

The PHY is now in mission mode. The outputs are controlled by the DFI bus inputs and must remain valid as long as the PHY is in an operational state. If training was run, the memory is left initialized and in the Self-Refresh state.

### 6.4.2 Example Timing Diagram for PHY Initialization operation

The following diagram illustrate the sequence for cold boot, initialization and a frequency change to the mission mode. This diagram does not include:

The execution of the DevInit/Training firmware, which would happen as part of step (G).

**Figure 6-8 Example Timing Diagram for PHY Initialization**

Note: PHYINIT PState order of execution is determined by PHYINIT userInput->FirstPstate. The FirstPstate value will be the last to be executed by PHYINIT. For example, for 4 PState's and FirstPstate=0, the order will be p1/p2/p3/p0 then boot into p0.

#### 6.4.3 Details of Example Timing Diagram of PHY Initialization

##### A: Initial Power Supply Ramp

- Supplies may stabilize in any order
- DfiClk, Reset, Reset\_async, WRSTN may be X
- BP\_PWROK may be 0 or 1.

##### B1: All Power Rails Up

- DfiClk, Reset, Reset\_async, WRSTN must be valid
- scan\_mode must be 0 and Reset/Reset\_async must be 1.

##### B2: PHY Reset Period

- TxBypassModeEn\*Ext=0, scan\_mode=0, Reset/Reset\_Async=1.
- APBCLK, DfiClk must follow reset sequence defined in diagram above.

##### B3: PHY Reset Exit

- APB/DFI inputs at valid reset-state
- Reset and Reset\_async input should be de-asserted low.
- If BP\_PWROK is low in step A, it should be asserted high now.

##### C, D, E, F, G, H : Pre DFI-Init (and Training)

- PHY FW/CSRs must be loaded

- PHY Training may be invoked
- PHY Training will set MEMRESET\_L = 1

I, J: DFI-initialization @ Requested Frequency

- CSR bit ClrPORMemReset should be programmed 1'b1 and followed by 1'b0 to switch the self initialized LATCH state.
- PHY will assert dfi\_init\_complete when ready

K: Mission Mode operation

DFI interface sets SDRAM pin values.

## 6.5 Power Management

The PHY power states can be grouped into two categories: Active (Mission Mode) and Standby.

### 6.5.1 Active / Mission Mode PHY Power States

The PHY supports 4 independent active power states (P0, P1, P2, P3). The power states have the following restrictions:

- Each PState must have a predefined dfi\_freq\_ratio.
- Each PState supports a maximum transfer rate specific to the supported PHY and voltage limitations.
- Each PState can be loaded in any DRAM FSP, for example, PHY PState is independent of DRAM FSP state.
- All PStates must support same page size bank (8B mode or 16B/BG mode) architecture.

Active power states are entered by a DFI Initialization or Frequency Change sequence, using the dfi\_init\_start and dfi\_frequency[4:0]/dfi\_freq\_ratio[1:0] interface.



See section “DRAM State During Frequency and Lower Power Changes” on page 175 for details of DRAM state requirement before initiating frequency change requests to enter an active state.

### 6.5.2 PHY Low Power and Standby States

The PHY provides a mix of low power states with varying exit latencies so that memory controller / SoC can dynamically take advantage of memory system idle windows.

**Table 6-2**    **PHY Low Power State Modes**

Low Power State	PHY Exit Latency	PHY State	DRAM Commands available	Enter /Exit Mechanism
IDLE	0	DFI LP Data Request asserted DATA BYTES stopped Receivers are in stand-by	All commands	DFI Command Interface
DFI_LPDATA_ACK	tlp_data_wak eup <sup>5</sup>	DATA BYTES stopped Receivers are in stand-by	All commands (available other than data-bearing transactions).	DFI_LPDATA_ACK
DFI_LPCTRL_ACK	tlp_ctrl_wake up <sup>5</sup>	Chip selects inactive AC signals tristated/inactive MEMCLK toggling	DES or Maintain Power-down or Maintain Self-Refresh	DFI_LPCTRL_ACK

Low Power State	PHY Exit Latency	PHY State	DRAM Commands available	Enter /Exit Mechanism
DFI_LPCTRL_ACK +DFI_LPDATA_ACK +DRAMCLKSTOP	tlp_wakeup <sup>5</sup>	DATA BYTES stopped Receivers are in stand-by Chip selects inactive AC signals tristated/inactive MEMCLK stopped DATA BYTES Receivers are in a light PowerDown state <sup>4 7</sup> DFI Clock to AC and DBYTE Hard Macros stopped <sup>6 7</sup> DFI Clock to PUB_AC and PUB_DX stopped <sup>6 7</sup>	Maintain Power-down or Maintain Self-Refresh Power-down	DFI_LPCTRL_ACK
LP2+DFICLKSTOP (PHY Fast Standby)	tLP2 <sup>1 2</sup>	DATA BYTES stopped Receivers powered-down Chip selects inactive AC signals tristated/inactive MEMCLK stopped DFICLK stopped ZQ/LCDL Calibration off PLL in Standby	Maintain Self-Refresh Power-down	DFI status interface
PHY IO Retention	tLP3 <sup>1 3</sup>	CS* driven to zero, MEMRESET_L driven to high. All other Pins are tristated. VDD is off	Maintain Self-Refresh Power-down	DFI Status Interface

Notes:

1. tLP2 and tLP3 are defined in “[DFI Status Interface Latencies](#)” on page 177
2. tLP2 is tinit\_complete for Relock+Retrain. Refer to “[DFI Status Interface Latencies](#)” on page 177
3. tLP3 is tinit\_complete for Cold Boot/Retention Exit. Refer to “[DFI Status Interface Latencies](#)” on page 177.
4. This is available by setting EnLpDqPowerDownLight in phyinit
5. Refer to “[DFI Low Power Interface](#)” on page 316
6. This does not happen immediately and may not happen if the PHY is not in this state long enough
7. Depends on the state of csrLpDqPhaseDisable, and is only available with Maximum power savings.

### 6.5.3 DFI Low Power Interface (DFI LP)

The DFI Low Power Interface can be used to perform clock gating when there is a sufficient period of time between traffic through the PHY.

Entering and exiting the DFI LP state is performed according to the DFI LP protocol and may be performed while the DFI interface is active (for example, dfi\_init\_complete asserted \*and\* no ongoing frequency change handshake).

Prior to performing frequency or power state changes (including LP2/LP3), the `dfi_lp_{data,ctrl}_req` must be de-asserted, since this interface must be inactive prior to frequency changes through `dfi_init_start`. This clean exit from DFI LP prior to frequency or power state changes ensures safe and well defined transitions and behaviors between power-states, while also preventing possible violations of the `dfi_lp` handshakes.



**Note** The LP2/LP3 power states provides superior power savings to that of the DFI LP alone

In two-channel operation, the DFI Low Power Interfaces operate independently of one another. If the PHY acknowledges a “`dfi_lp_data_req`” request, the PHY performs the following power savings actions:

- Stop the clocks to the data bytes.
- Place all DQ/DQS receivers in standby/powerdown



**Note** `dfi_lp_data_req` may be entered anytime there are no in-progress dram read/write transactions.

If the PHY acknowledges a “`dfi_lp_ctrl_req`” request, the PHY performs the following power savings actions:

- Stop the clocks to the address/command transmitters.



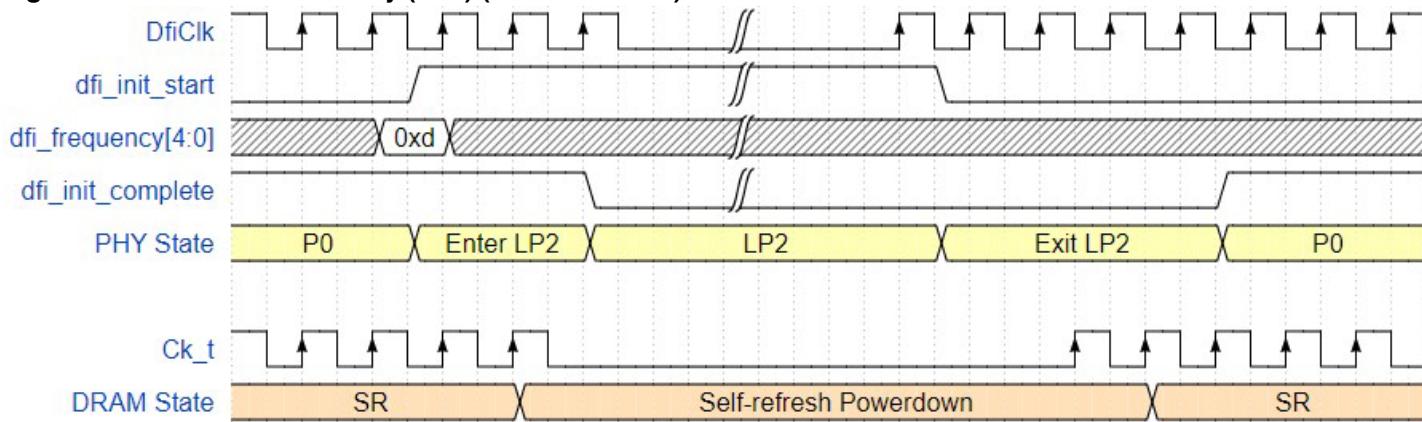
- The MEMCLK outputs to the dram will still be running (unless the memory controller explicitly asserts the `dfi_dram_clk_disable` pins – for additional power-savings)
- For very short assertions of `dfi_lp_ctrl_req` or `dfi_lp_data_req`, some power-saving measures may not be activated.

## 6.5.4 PHY Fast Standby (LP2)

PHY Fast Standby (LP2) allows an SOC to save the maximum power while the DRAM is in Self-Refresh and maintain a reasonable exit latency. MC can enter this state by leveraging the Frequency Change Protocol and DFI Status interface. [Figure 6-9](#) on page [160](#) shows a transaction where the source and destination state are equal. The following list shows the requirements of a PHY Fast Standby (LP2) transaction:

By entering LP2, the controller may clock gate the entire PHY.

- Refer to table “[dfi\\_frequency Encoding](#)” on page [173](#)
- LP2 exit latency (`tLP2`) is dictated by `tinit_complete` for Relock+Retrain for the given PState.

**Figure 6-9** **PHY Fast Standby (LP2) (LPDDR5/5X/4x)****Note**

During PHY LP2, the PHY impedance calibration state machine is paused and calibrated values from initial training are loaded. If the residency in LP2 is long enough that the system temperature has varied by greater than 20 C, it is recommended for the controller to wait an additional 20  $\mu$ s after asserting edge of dfi\_init\_complete - before issuing the commands to DRAMs. DFI sideband event must be used to propagate the new code to the IO. If the temperature change while in LP2 has not exceeded the threshold, the extra wait is not required.

### 6.5.5 LP3/IO Retention (LP3)

The LP3/IO Retention power state is a non-operational low power state. The standby power state has the following properties:

- This is the lowest possible power state for the PHY.
- In this power state of the DRAM is defined by [Table 6-6](#) on page 176. For further detail see “[DRAM State During Frequency and Lower Power Changes](#)” on page 175.
- The BP\_PWROK input is used to indicate entry into this state to the PHY.
- BP\_PWROK must be in always on VDD2H power domain.
- The core VDD, VAA and VDDQ power supplies can be powered down in this state. If the core VDD power supply is shut off, power consumption is predominantly determined by the device leakage on VDD2H and the number of active CS lanes.
- VDD2H must remain valid for the value of the pins to be preserved.
- The Standby power state is entered via a Frequency Change sequence, using the dfi\_init\_start and dfi\_frequency[4:0]/dfi\_freq\_ratio[1:0] interface.

Refer to table “[dfi\\_frequency Encoding](#)” on page 173.

**Note**

When entering or exiting LP3/IO Retention, the DFI interface must be held in a valid state until BP\_PWROK is set to 0. While in LP3/IO Retention, the DRAM signals required for proper IO Retention will maintain the appropriate state (including CS and MEMRESET\_L pins). The Memory Controller may ignore the dfi\_phyupd\_req signals during LP3.



**Note** Prior to the LP3/IO Retention Enter procedure, the values of the PHY configuration and trained registers must be saved in order to restore after LP3/IO Retention Exit. The next sections describe saving registers, entering and exiting retention state.

For proper retention operation, BP\_PWROK must be held at 0 while VDD is above the threshold level during VDD ramp up or ramp down.

The recommended process is to drive BP\_PWROK directly from an always-on VDD2H power domain.

#### 6.5.5.1 LP3/IO Retention Save Registers

To correctly exit retention, the values of PHY registers, including those trained by firmware or manually programmed, must be saved prior to entering mission mode. All writable PHY registers can be saved, or, if the configuration is known, PhyInit can be used to generate an optimized list.



**Note** The number of registers that must be saved prior to LP3/IO Retention entry and restored post LP3/IO Retention Exit varies based upon the DDR standard. PhyInit can be used to generate an initialization sequence that automatically includes saving retention registers based on PHY configuration and protocol. Refer to the DDRn PhyInit Application Note for more information.

The sequence generated by PhyInit for I/O Retention Save performs the following steps:

1. Write the MicroContMuxSel CSR to 0x0 to allow access to the internal CSRs.
2. Write the UcclkHclkEnables CSR to 0x07 to enable all the clocks so the reads can complete.
3. Read and save all the registers appropriate for the given protocol and training.
4. The register address/data pairs must be saved in some part of the system (such as external non-volatile storage).
5. Write the UcclkHclkEnables CSR to 0x4 to disable the appropriate clocks after all reads done.
6. Write the MicroContMuxSel CSR to 0x1 to isolate the internal CSRs during mission mode.

### 6.5.5.2 LP3 / IO Retention Enter

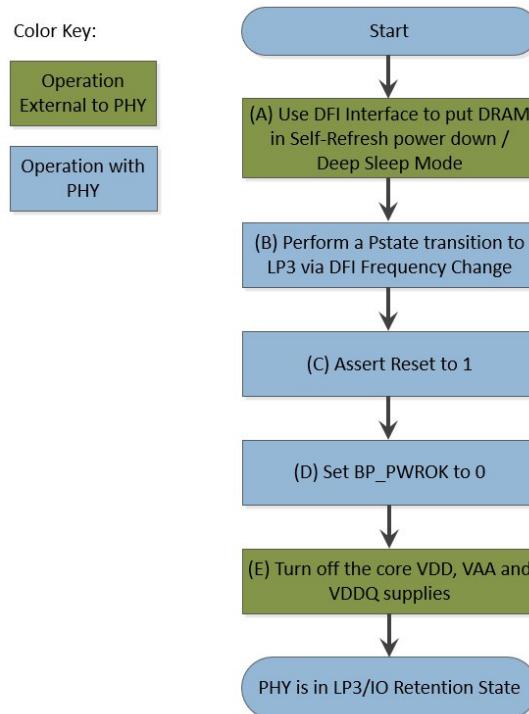
Transitioning to the LP3/IO Retention power state has a few more steps than the active power state transitions.



**Note** Prior to the LP3/IO Retention Enter procedure, the values of the PHY configuration and trained registers must be saved as part of the PHY initialization process. Refer to “[LP3/IO Retention Save Registers](#)” on page [161](#).

The following is the full process to enter LP3/IO Retention:

**Figure 6-10 Retention Entry Sequence**

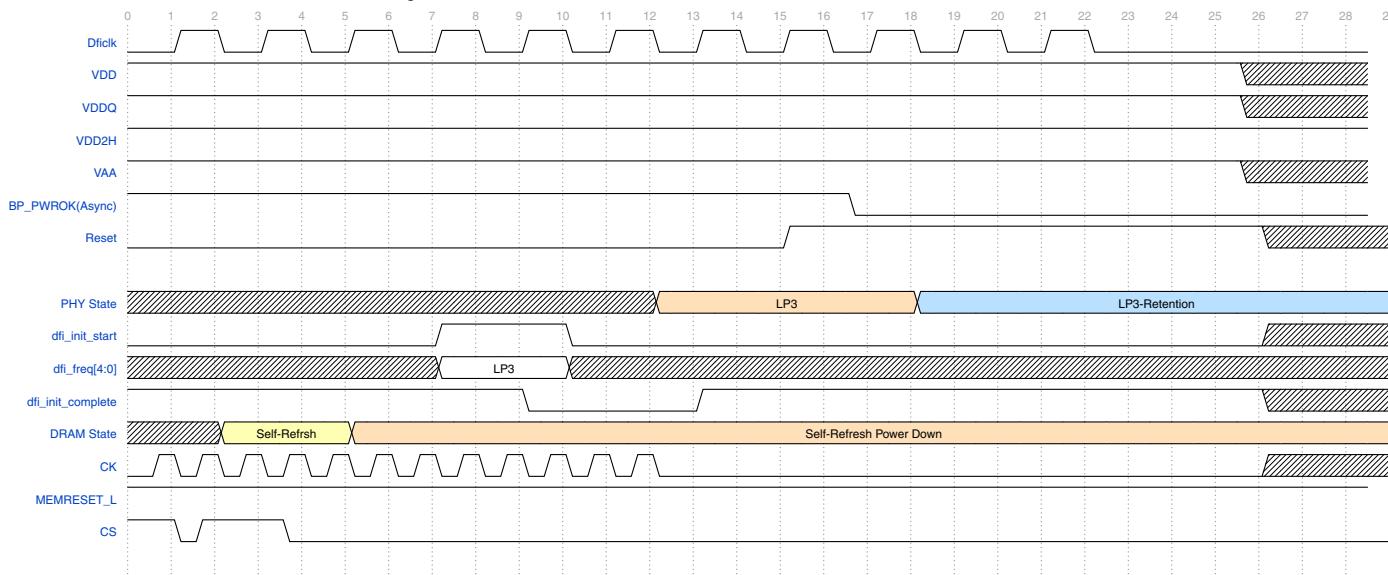


- (A) Use the DFI interface to put the DRAM into Self-Refresh Power Down or Deep Sleep Mode.
  - LPDDR5/5X:: if enabled:  
When background calibration mode is selected (MR28 ZQ Mode = 0), MC is required to program MR28 ZQ Stop = 1 before initiating a frequency change request that places the PHY in LP3 state when VDDQ is going to be powered off.  
When Command-based calibration mode is selected (MR28 ZQ Mode = 1), MC is required to keep MR28 ZQ Stop = 0 before asserting dfi\_init\_start = 1.
- (B) Transition the PHY to the LP3 state by using a DFI Frequency Change operation.
  - MEMRESET\_L will be protected and held at 1.
  - After dfi\_init\_complete transitions from 0 -> 1:
    - CS will be held at 0.

- MEMRESET\_L will be held at 1.
- All other output pins will be in un-driven (non deterministic) state.
- VDD must remain at valid level.
- (C) Assert Reset input to 1.
  - This will enable the data retention feature on the CS and MEMRESET\_L pins.
  - CS will be held at 0.
  - MEMRESET\_L will be held at 1.
  - All other output pins will be in un-driven (non deterministic) state.
  - Remaining PHY input signals must be valid, inactive states for at least 10ns after the BP\_PWROK asserts 1 -> 0.
  - The PHY input clocks are not required to toggle when BP\_PWROK=0.
- (D) Set the BP\_PWROK signal to 0.
  - This will enable the data retention feature on the CS and MEMRESET\_L pins.
  - CS will be held at 0.
  - MEMRESET\_L will be held at 1.
  - All other output pins will be in un-driven (non deterministic) state.
  - Remaining PHY input signals must be valid, inactive states for at least 10ns after the BP\_PWROK asserts 1 -> 0.
  - The PHY input clocks are not required to toggle when BP\_PWROK=0.
- (E) Turn off the core VDD, VAA and VDDQ supplies.
  - These supplies can be turned off in any order.
  - The VDD2H power supply must remain active during LP3/IO Retention.

This step is optional. Power supplies do not have to be powered down (turned off) at the end of LP3/IO Retention Enter. However, leaving the power supplies active does not provide the best possible power savings.

All PHY input values are now don't cares.

**Figure 6-11 LP3/IO Retention Entry**

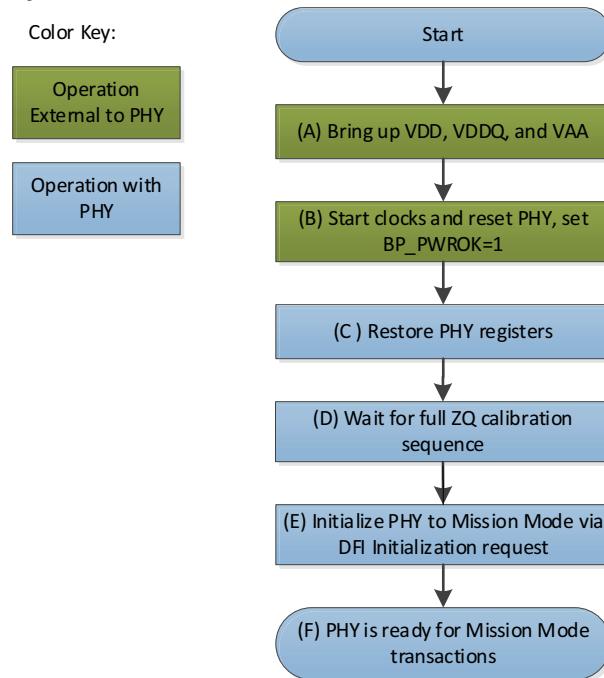
### 6.5.5.3 LP3 / IO Retention Exit

The process to exit LP3/IO Retention is a simplified compared to the initialization sequence. PhyInit can be used to automatically generate a detailed retention exit sequence for a given configuration and DRAM protocol. See “I/O Retention Save/Restore” section in the PhyInit Application note for details. Following section describes the sequence at a high level.



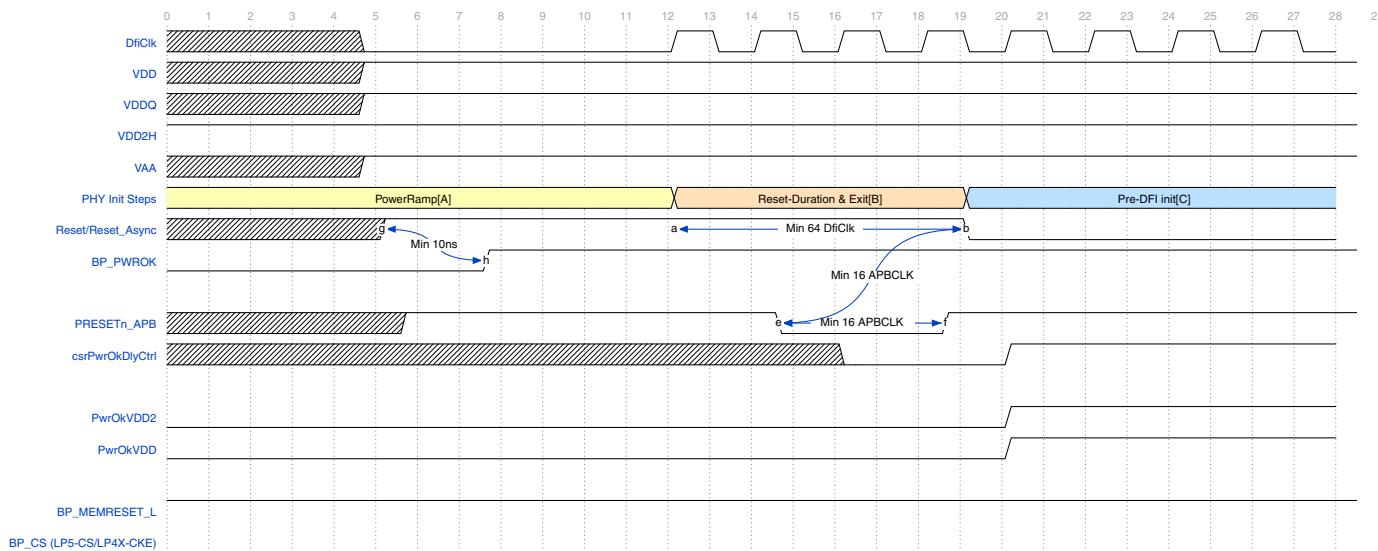
Prior to this procedure, the values of the PHY configuration and trained registers must have been saved where they can be used as part of this process. The register save process should be executed between steps I and J of “[PHY Initialization Sequence](#)” on section “[\(I\) Load PHY Init Engine Image](#)” on page 154 and “[\(J\) Initialize the PHY to Mission Mode through DFI Initialization](#)” on page 154. If the registers are not saved prior to the LP3/IO Retention Enter operation, then the full initialization procedure must be performed and the contents of the external DRAM is not guaranteed. For more information on the full initialization procedure, see “[PHY Initialization](#)” on page 151.

The exit LP3/IO Retention steps are described in the following process (see the procedure in [Figure 6-12](#) on page 165).

**Figure 6-12 IO Retention Exit Sequence**

1. The VDD2H power supply will already be active; restore the core VDD, VAA and VDDQ power supplies similar to Step A in “[PHY Initialization](#)” on page 151.
2. Start clocks and reset the PHY similar to Step B in “[PHY Initialization Sequence](#)” on page 151.
  - ❑ Assert BP\_PWROK input.
    - BP\_PWROK input should be asserted as shown in [Figure 6-8](#) on page 155.
    - ❑ The csrPwrOkDly will be asserted by phyinit restore\_sequence.
    - ❑ The {TxBypassModeEn\*, WRSTN} signals may be undefined at VDD power-on, but must be driven LOW at least 10ns prior to the asserting edge of BP\_PWROK.
3. Restore all PHY registers:  
Set up the PHY to write registers:
  - a. Write the MicroContMuxSel CSR to 0x0 to allow access to the internal CSRs
  - b. Write the UcclkHclkEnables CSR to 0x07 to enable all the clocks so the reads can complete
  - c. Write ZCalReset to 0x1

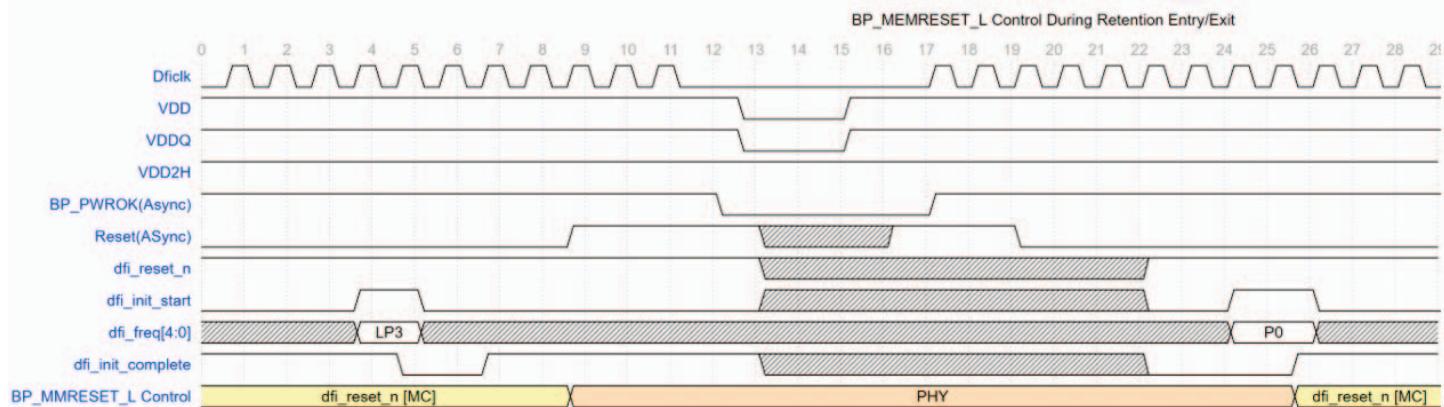
- d. Issue register writes restoring values. This includes:
  - i. Initialize the PHY configuration by writing registers using the dwc\_ddrphy\_phyinit\_restore\_sequence() function. See section "[LP3/IO Retention Save Registers](#)" on page [161](#) for details.
  - ii. Write any user custom registers added in the initialization sequence before training execution as part of dwc\_ddrphy\_phyinit\_userCustom\_customPre() functions of PhyInit.
  - iii. Write any user custom registers added in the initialization sequence after training execution as part of dwc\_ddrphy\_phyinit\_userCustom\_customPost() functions of PhyInit.
  - iv. Load the PHY Initialization Engine image. This includes registers part of Step I of "[PHY Initialization Sequence](#)" on page [151](#).
  - v. ACSM and PIE SRAM
  - vi. Pulse TtcfControl CSR
4. Wait for full ZQ calibration sequence
  - i. Host writes csrPstate that corresponds to the DfiClk frequency
  - ii. Host triggers the ZCAL FSM by writing csrZCalReset=0, followed by csZCalRun=1.
  - iii. Host waits for completions of ZQ Calibration (47us)
  - iv. Host sets csrZCalReset=1
5. Initialize the PHY to mission mode through a DFI initialization request. This step is similar to Step J "[PHY Initialization Sequence](#)" on page [151](#).
  - a. Write the UcclkHclkEnables CSR to disable the appropriate clocks after all reads done: write the UcclkHclkEnables to 0x4
  - b. Write the MicroContMuxSel CSR to 0x1 to isolate the internal CSRs during mission mode
  - c. MC should drive valid and stable dfi\_reset\_n/dfi\_cs along with dfi\_init\_start.
  - d. PHY will control MEMRESET\_L /CS pad status while dfi\_init\_complete is low.
  - e. LPDDR5/5X:: if enabled:  
During 1st dfi\_init\_start and LP3 exit sequence, PHY will program MR28 with default values when Background ZQ calibration mode is selected. It is allowed for controller to change MR28 values before starting mission mode operation. However, MR28 ZQ Mode must be same value as Phyinit Userinput ZqMode before entering LP3.
6. PHY is ready for mission-mode transactions

**Figure 6-13 LP3 IO Retention Exit**

#### 6.5.5.4 Protection of BP\_MEMRESET\_L During Retention Entry/Exit

During mission mode BP\_MEMRESET\_L signal is controlled by MC through dfi\_reset\_n on VDD domain. Since the state of this signal may become unknown during retention, special care must be taken to ensure BP\_MEMRESET\_L is not de-asserted.

The following figure shows the state of BP\_MEMRESET\_L during retention entry and exit.

**Figure 6-14 BP\_MEMRESET\_L Control During Retention Entry/Exit**

The PHY (PUB + POR circuit) asserts BP\_MMRESET\_L when in control. The memory controller must ensure dfi\_reset\_n is asserted during yellow periods in diagram above. Alternatively MemResetL register can be used to override dfi\_reset\_n.

#### 6.5.5.5 Register Restore Timing

To save the trained state, the values of all the trained CSRs must be read after the training firmware is run and before the PHY is put into mission mode (between steps I and J shown in [Figure 6-14](#) on page 167).

To perform the register save process, “[LP3/IO Retention Save Registers](#)” on page 161:

The number of registers that must be saved prior to LP3/IO Retention entry and restored post LP3/IO Retention Exit varies based upon the DDR standard and the training steps run.

Depending on PHY configuration, for example, Number of PStates, Dbytes, Ranks, etc. different number of registers must be restored. The exact list for a given configuration is generated by PhyInit as part of the exit retention sequence. Refer to PhyInit output.txt files; example below:

```
// [dwc_ddrphy_phyinit_regInterface] Total number of LP3 IO Retention (S3) CSR reads = 4460 (includes ACSM/PIE SRAM CSRs)
// [dwc_ddrphy_phyinit_regInterface] Number of LP3 IO Retention (S3) ACSM SRAM reads = 1896
// [dwc_ddrphy_phyinit_regInterface] Number of LP3 IO Retention (S3) PIE SRAM reads = 1524
// [dwc_ddrphy_phyinit_regInterface] Number of LP3 IO Retention (S3) PHY CSR reads = 1040 (excludes ACSM/PIE SRAM CSRs)
```

Following table provides some examples for reference based on specified conditions.

**Table 6-3 Number of Restore Registers**

Config	Number of APB reads (Nreg)
Dual Channel (32 bit) PHY, NumPState=1	4500
Dual Channel (32 bit) PHY, NumPState=2	5200
Dual Channel (32 bit) PHY, NumPState=4	7500
Single Channel(16 bit)PHY, NumPState=4	5800

The restore time can be given by:

$$t_{\text{Restore}} (\text{num of DfiClks}) = N_{\text{reg}} * 4 + t_{\text{init\_complete}}$$

## 6.6 DFI Status Interface

The DFI status interface is used by the MC control to change the state of the PHY. It is used to:

- Initialize the PHY to a PState
- Enter/Exit Low power states
- Change the PState
- Perform retraining

The interface follows the DFI spec for Initialization and Frequency Change protocols. See the official DFI for more information on the protocol. This section describes additional details and requirement of the PHY.

DFI status interactions are handled by the PHY Initialization Engine (PIE) described in details in “[PHY Initialization](#)” on page [151](#). This HW state machine ensures that Power state transitions, PLL-lock, LCDL-lock, and other sequence of events occur correctly.

### 6.6.1 DFI Initialization

The DFI initialization protocol is used for the first transition of dfi\_init\_start. The protocol is as follow:

1. d<sub>i</sub>f<sub>i</sub>\_init\_start must start at 0.
2. Set the value of d<sub>i</sub>f<sub>i</sub>\_frequency[4:0], d<sub>i</sub>f<sub>i</sub>\_freq\_fsp[1:0] and d<sub>i</sub>f<sub>i</sub>\_freq\_ratio[1:0] to the appropriate value for the desired target power state. It is legal to use any active power state as the destination power state. The destination power state must have been trained using the correct frequency.
3. Set d<sub>i</sub>f<sub>i</sub>\_init\_start to 1.
4. Wait for d<sub>i</sub>f<sub>i</sub>\_init\_complete to be asserted
5. Before the first frequency change operation, d<sub>i</sub>f<sub>i</sub>\_init\_start must be set to 0. The timing of this operation is not critical, but must happen at least 16 cycles before the first frequency change

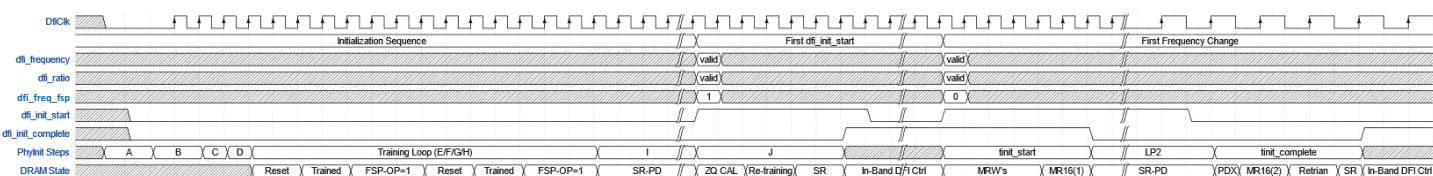
The figures below show the first DFI transaction after training. The PIE will send the following to commands to the DRAM on first d<sub>i</sub>f<sub>i</sub>\_init\_start/complete during step 4:

LPDDR5/5x: If DsableFspOp=0, PHY will send FSP-OP/FSP-WR

DRAM ZQ Cal. (LPDDR5: only if background ZQ is disabled)

Perform retraining if required by target PState.

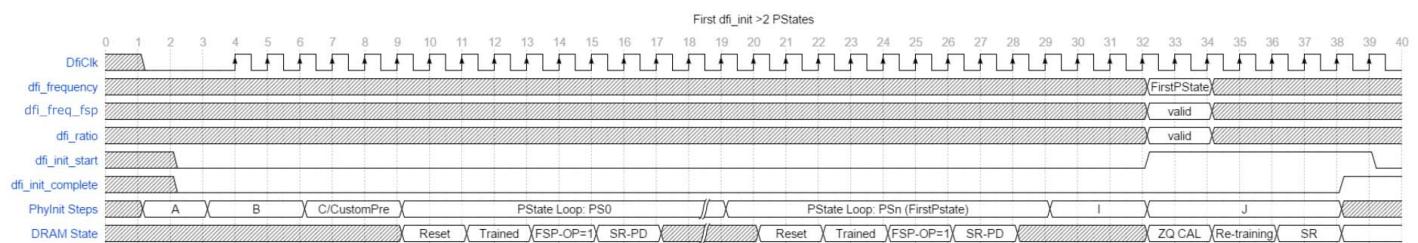
**Figure 6-15 Initialization, First and Second d<sub>i</sub>f<sub>i</sub>\_init\_start for LPDDR5/5x**



1. MR16(1) sets FSP-OP=d<sub>i</sub>f<sub>i</sub>\_freq\_fsp, FSP-WR= !d<sub>i</sub>f<sub>i</sub>\_freq\_fsp, VRCG=high.
2. MR16(2) sets FSP-OP=d<sub>i</sub>f<sub>i</sub>\_freq\_fsp, FSP-WR= !d<sub>i</sub>f<sub>i</sub>\_freq\_fsp, VRCG=low.



**Note** For both userInputBasic.DisableFspOp=0 and 1, d<sub>i</sub>f<sub>i</sub>\_freq\_fsp must always be valid during all DFI Status requests.

**Figure 6-16 Initialization and First dfi\_init\_start DisableFspOp=0**

**Note** In two channel mode, the PHY asserts and de-asserts dfi0\_init\_complete and dfi1\_init\_complete simultaneously. The PHY samples DFI status interface signals on rising edge of DfiClk.

On a 0 to 1 transition of sampled dfi\_init\_start, the PHY considers samples of latter signals valid, representing the target PState and the transition method. The signals are ignored at all other times.

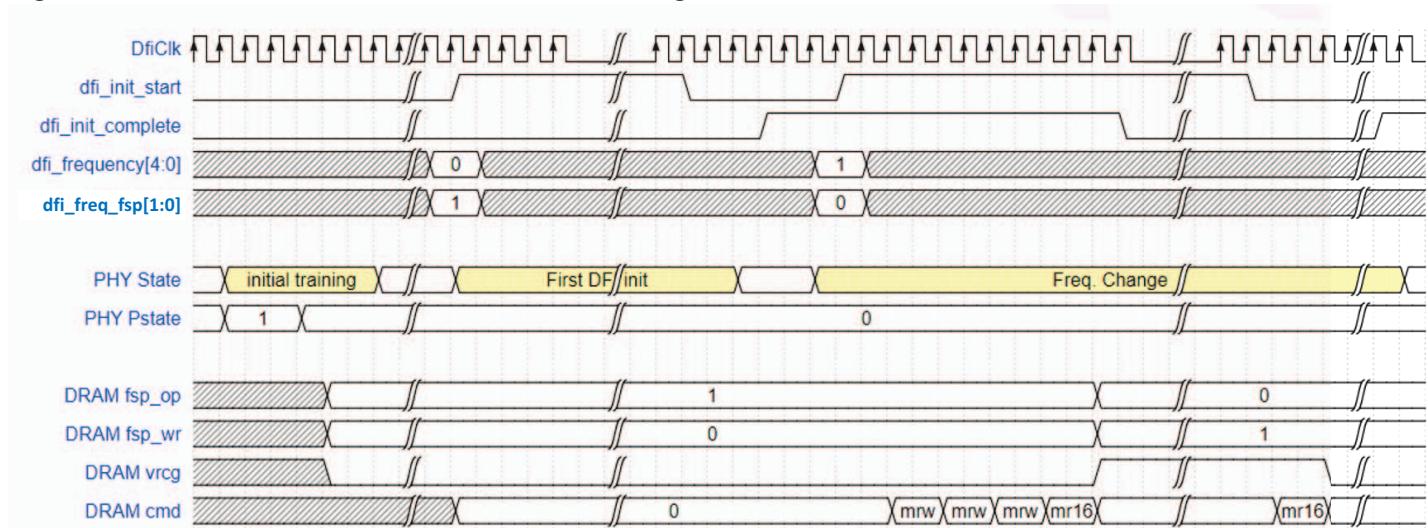
In dfi frequency change sequence, PHYINIT UserInput DisableFspOp=0 is default and recommended setting.

However, MC has option to set MRW and FSP-OP for destination PState by using PHYINIT UserInput DisableFspOp = 1 as below:

- In LPDDR4X mode:
  - When DisableFspOp=0, PHY will send DRAM MRW for destination PState and will set MR13 FSP-OP based on dfi\_freq\_fsp input.
  - When DisableFspOp=1, PHY will not send DRAM MRW for destination PState and will not set MR13 FSP-OP. MC must send DRAM MRW as well as MR13 FSP-OP.
- In LPDDR5 mode:
  - When DisableFspOp=0, PHY will send DRAM MRW for destination and will set MR16 FSP-OP based on dfi\_freq\_fsp input.
  - When DisableFspOp=1, PHY will not send DRAM MRW for destination. However, PHY will set MR16 FSP-OP based on dfi\_freq\_fsp input.

### 6.6.1.1 State of DRAM on first initialization

The state of the DRAM after initial training is always set FSP=1. Thus on cold boot exit, dfi\_freq\_fsp must always be driven to 0x1. In order to optimize the state of DRAM, PhyInit variable userInputBasic.FirstPstate must be set to match the encoding used for dfi\_frequency[4:0].

**Figure 6-17 PHY PState and DRAM FSP State During Initialization when FirstPState=0**

Note: List of MRW is protocol (LPDDR5/5X or LPDDR4X) dependent. MR16 shown above indicates LPDDR5/5X FSP-OP switch. For LPDDR4X, MR13 is used for FSP-OP switch.

## 6.6.2 DFI Frequency Change

The DFI frequency change protocol is used for all transitions after the first transitions.

1. Set the value of DFI signals to the appropriate value for the desired target power state.
2. dfi\_frequency[4:0], dfi\_freq\_fsp[1:0] and dfi\_freq\_ratio[1:0]



LPDDR5x: If the encoding of {dfi\_frequency[4:0], dfi\_freq\_ratio[1:0]} represents a PState transition that required changing for DRAM FSP, then dfi\_freq\_fsp[0] must be flipped to select a different DRAM FSP for the target PState.

3. Set dfi\_init\_start to 1.
4. Wait for dfi\_init\_complete to be set to 0 (de-asserted) (tinit\_start).
5. Change the DfiClk frequency to the new desired frequency.
6. Once the clock is stable at the new frequency, set dfi\_init\_start=0.
7. Wait for dfi\_init\_complete to be asserted (tinit\_complete).

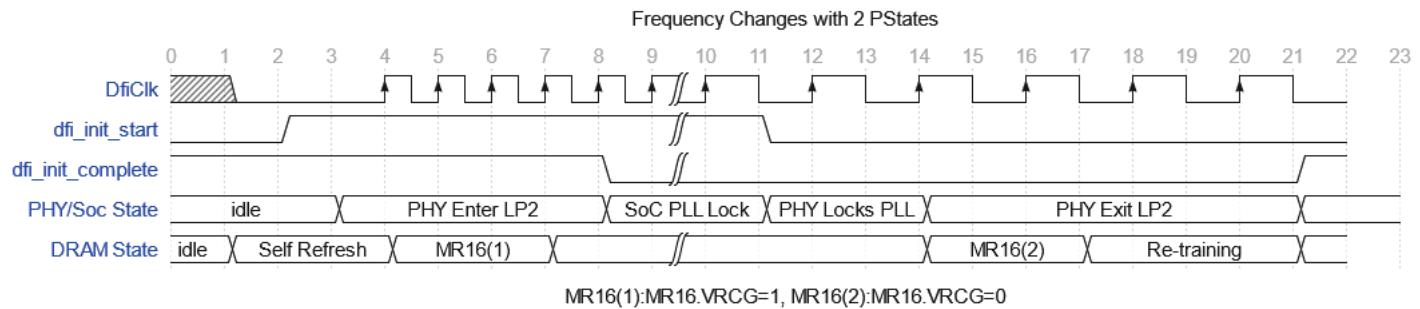
The PIE performs the following during step 3 (tinit\_start):

- PHY writes mode register values for the target DRAM FSP if the previously loaded FSP state was for a different PState.
- PHY performs MRW to trigger DRAM FSP-OP switch to target FSP if CA-ODT is off at source PState.
- Receivers should be powered down
- ZqCal FSM is paused
- PLL is moved to Standby mode.

The PIE performs the following during step 7 (tinit\_complete):

- Start re-locking the PLL if required and resume ZQCAl FSM
- Relock LCDLs
- Re-Initialize data path pipeline
- PHY performs MRW to trigger DRAM FSP-OP switch to target FSP. If CA-ODT is on at source PState.
- Perform periodic retraining, without waiting for ZQCAl completion.

**Figure 6-18 Subsequent Frequency Changes After First Two dfi\_init start requests, DisableFsp=0, LPDDR5/5x**



1. MR16(1) sets FSP-OP= dfi\_freq\_fsp, FSP-WR =!dfi\_freq\_fsp, VRCG=high.
2. MR16(2) sets FSP-OP= dfi\_freq\_fsp, FSP-WR =!dfi\_freq\_fsp, VRCG=low.



**Note** When only two PStates are defined, dfi\_freq\_fsp is implied however it must still be driven accordingly if userInputElementBasic.DisableFsp=0.

When scaling of PHY core VDD is supported, SOC is allowed to ramp VDD up or down during PState Change (relock+retrain) sequence (after dfi\_init\_complete transitions from 1 -> 0 and DfiClk/PllRefClk/PllBypClk Clock inputs have stopped).

### 6.6.3 PHY State Changes and dfi\_frequency Encoding

The PHY supports low-latency dynamic frequency changes through the in-band DFI frequency change protocol. The PHY maintains 4 independent HW register contexts.

Operating state switching is performed with following the DFI Frequency change protocol using the signals {dfi\_init\_start,dfi\_frequency[4:0],dfi\_freq\_ratio[1:0], dfi\_freq\_fsp, dfi\_init\_complete}. The following table shows the allowable values for PHY inputs in DFI status interface transitions.

dfi\_freq\_ratio must match the PhyInit userInputBasic.DfiFreqRatio setting for the corresponding PState.

**Table 6-4** **dfi\_frequency Encoding**

<b>dfi_frequency[4:0]</b>	<b>Description</b>	<b>Notes</b>
0x0	P0-P3, Relock+Retrain	2, 4, 7
0x1		
0x2		
0x3		
0x4	P0-P3 Fast Relock + Retrain	1
0x5		
0x6		
0x7		
0x8-0xb	Reserved	
0xc	Retrain Only (Remain at Current PState)	1, 3, 4, 7
0xd	LP2 Enter/Exit (Remain at Current PState) (Skip Retraining and keep previous timing results)	2, 7
0xe	RFU	
0xf	RFU	
0x10	P0-P3 Relock Only (Skip Retraining and keep previous timing results)	2, 4, 6
0x11		
0x12		
0x13		
0x14-0x1c	RFU	
0x1d	RFU	
0x1e	RFU	
0x1f	Enter LP3 (PHY Deep Sleep/Retention)	5

1. During Fast Relock+ Retrain frequency change, the PHY changes datarate by keeping the PLL locked and changing PLL post dividers. Thus DFICLK must run continuously (not be gated and/or change frequency) when { dfi\_init\_start=1, dfi\_init\_complete=0}. To reduce link downtime, it is recommended for MC to de-assert dfi\_init\_start as soon as it sees dfi\_init\_complete=0. Fast Relock+ Retrain encoding is used to switch between a PState with data rate N to another PState with data rate N/2 or vice versa where PLL is locked for data rate N.
2. If PllBypass is enabled via PhyInit, Pll Relock is always skipped. For any Data rate under DDR3200 PLL can be bypassed or not bypassed. For any data rate under DDR667 PLL must be bypassed.

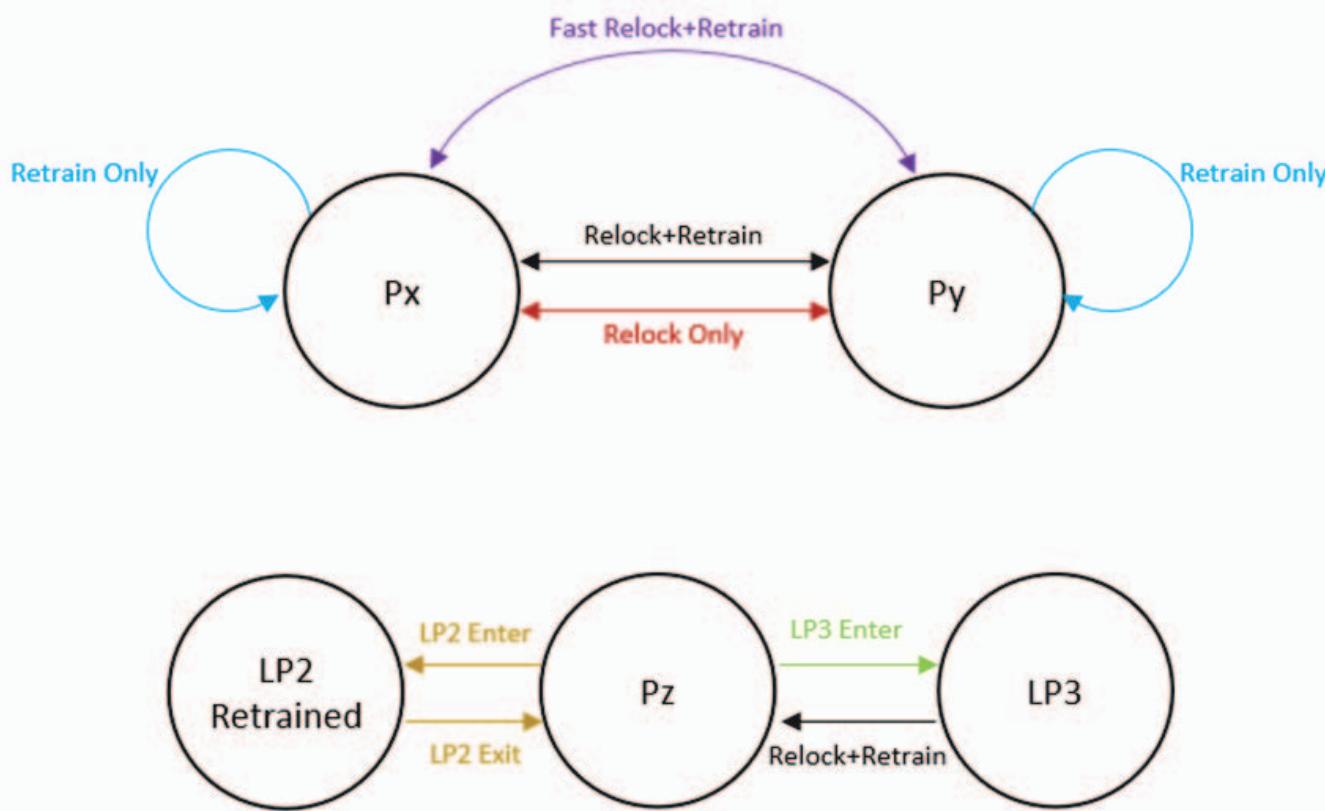
3. Only valid for datarates equal or above DDR1600, Retraining is not supported at datarates below DDR1600.
4. DFICLK may be gated and/or change frequency when {dfi\_init\_start=1,dfi\_init\_complete=0}
5. LP3 Exit should be always followed by Relock+Retrain. Other frequency transitions are deemed invalid and will result unexpected behavior. In addition PHY must Enter and Exit the same PState on LP3.
6. When Relock Only frequency Change is executed, no compensation of tWCK2DQI, tWCK2DQO (LPDDR5/5Xx), tDQS2DQ, or tDQSCK (LPDDR4X), or tPHY\_DQS2DQ drift is applied and initial training results are used. This encoding should only be used when drift conditions are the same as initial training or when drift compensation is not required, for example, Date rates < 1600 Mbps
7. ZQ update is skipped when data rate <= 3200 Mbps.



- If PllBypass is enabled for a PState via PhyInit, PLL Relock is always skipped.
- dfi\_freq\_ratio must match the PhyInit userInputBasic.DfiFreqRatio setting for the corresponding PState.
- The "Fast Relock" option in the PUB databook is a separate feature to the PLL's fast relock mode.

The diagram below shows valid transitions between PHY PStates. In order to support proper transition from one PState over another, MUX implementation which supports PState selection need to be glitchless.

**Figure 6-19 PHY Transition Diagrams**





LPDDR5/5X/4X: PHY must enter/exit LP3 from the same PState. For example Px --> LP3 must be followed by LP3 --> Px. This restriction is in place due to nature of FSP operation during to Frequency change. See “[Special Consideration on dfi\\_freq\\_fsp \(LPDDR5/5x\)](#)” on page [175](#).

The table below shows which sequences are part of which transition type.

**Table 6-5 Frequency Change Sequence Breakdown**

	First dfi_init				Retrain+Relock	Relock	Retraining	Fast Relock + Retrain	Subsequent dfi_init		
	Retrain+Relock	Relock	Retraining	Fast Relock + Retrain					Retrain+Relock	Relock	Retraining / PMI
PHY ZQ Cal Update <sup>a</sup>	X	INVALID	INVALID	INVALID	X	X	X	X	X	X	X
PLL Lock	X				X	X				X	X
LCDL Calibration	X				X	X		X	X	X	
PCLK DCA Calibration	X				X	X		X	X	X	
PtrlInit Sequence	X				X	X		X	X	X	
DRAM MPC ZQ CAL	X										X
Retraining	X				X		X	X			X

a. For PHY ZQ calibration management details, refer to “[PHY ZQ Calibration Management by PIE Sequence](#)” on page [227](#).

#### 6.6.4 Special Consideration on dfi\_freq\_fsp (LPDDR5/5x)

The PHY frequency change protocol supports only 2 DRAM FSP's. For optimal performance, PHY always leaves FSP\_WR to the non-active FSP. This allows to reduce the number of MRW to minimize tinit\_start. For correct operation MC must always flip dfi\_freq\_fsp when dfi\_frequency encoding requires a change in DRAM FSP state. The frequency change to same PState or re-train only does not require dfi\_freq\_fsp to toggle.

On retention entry/exit, PHY cannot change DRAM FSP. Thus DRAM FSP and PHY Pstate must remain the same on retention entry/exit. On cold boot, since training firmware leaves DRAM in FSP=1, dfi\_freq\_fsp must be driven to 0x1. Phyinit userInputBasic.FirstPstate variable can be used to control the initialization sequence to achieve the desired boot Pstate. This Pstate will always load in DRAM FSP=1.

#### 6.6.5 DRAM State During Frequency and Lower Power Changes

Following table describes in detail the DRAM states which MC can place the DRAM before initiating a DFI Status interface (dfi\_init\_start=0x1) or acknowledging a DFI phymaster request (dfi\_phymstr\_ack=0x1). It also shows the DRAM state when PHY returns the DRAM to the MC before assertion of dfi\_init\_complete=0x1 or dfi\_phymstr\_req=0x0.

**Table 6-6 DRAM State During DFI Status Update Requests**

	MC Places DRAM in ... <sup>1</sup>	PHY hands over DRAM in... <sup>2</sup>	DRAM State while PHY in Low Power State (waits for falling edge of dfi_init_start)
1st dfi_init_start (boot)	LPDDR5/5X: SR + PD LPDDR4X: SR + PD	LPDDR5/5X: SR LPDDR4X: SR + PD	-
Retention Entry	LPDDR5/5X: PDE or SRE+PD or SRE+DSM <sup>4</sup>  LPDDR4X: SR+PD	LPDDR5/5X: (no change)  LPDDR4X: (no change) Input Clock Stop CKE LOW CK_c/CK_t tristated.	LPDDR5/5X: CK_c/CK_t tristated. CS_n driven low CA tristated  LPDDR4X: (no change) Input Clock Stop CKE LOW CK_c/CK_t tristated. CA/CS tristated
Retention Exit	-	LPDDR5/5X : SR LPDDR4X : SR	-
LP2 Enter/Exit <sup>3</sup>	LPDDR5/5X: SR  LPDDR4X : SR	LPDDR5/5X : SR  LPDDR4X : SR	LPDDR5/5X: SR+PD CK_c/CK_t tristated.  LPDDR4X: SR+PD (Input Clock Stop CKE LOW) CK_c/CK_t tristated.
Relock Only <sup>6</sup> Relock + Retrain Fast Relock + Retrain	LPDDR5/5X: SR  LPDDR4X : SR <sup>3</sup>	LPDDR5/5X: No Change  LPDDR4X : SR <sup>3</sup>	LPDDR5/5X: SR + PD CK_c/CK_t tristated.  LPDDR4X: SR + PD (Input Clock Stop CKE LOW) CK_c/CK_t tristated.
DFI PHYMASTER Req/Ack Retrain Only	LPDDR5/5X: SR LPDDR4X : SR	LPDDR5/5X: SR LPDDR4X : SR	LPDDR5/5X: SR LPDDR4X : SR

- The DRAM must be in one of mentioned states on assertion of dfi\_init\_start or dfi\_phymstr\_ack, meeting all JEDEC timing requirement from previous command.
  - When Training Firmware is used, Firmware will put DRAM in SR +PD. Refer to Training Firmware Application Note section 5.19 End of Training.
  - When Training is skipped (for simulation), SOC environment will put DRAM in SR +PD before

controller asserts 1st dfi\_init\_start.

2. DRAM will be in this state on assertion of dfi\_init\_complete meeting all JEDEC timing requirement from previous command.
3. During this transition when PHY enters LP2 power state it paces the DRAM in Power Down state:
  - a. LPDDR4X: PHY places DRAM in SR+PD by setting CKE=0 before de-assertion of dfi\_init\_complete.
  - b. LPDDR5: PHY places DRAM in SR+PDE by issuing PDE command before de-assertion of dfi\_init\_complete.
4. PHY does not send PDE commands in LPDDR5 during retention entry sequence. DRAM state at the end of Retention entry sequence will be the same as what MC had placed the DRAM before initiating retention entry sequence. This gives the MC the freedom to choose any of the states list in the second column.
5. MC should drive valid and stable dfi\_reset\_n/dfi\_cs along with dfi\_init\_start
6. When Relock Only frequency Change is executed, no compensation of tDQSDQ or tDQSCK or tPHY\_DQS2DQ drift is applied and initial training results are used. This encoding should only be used when drift conditions are the same as initial training or when drift compensation is not required, for example, Date rates < 1600 Mbps.
7. If source and destination PStates are different, DRAM must be placed in SREF. If they are the same DRAM can be placed in SR.

### 6.6.6 DFI Status Interface Latencies

The following table and formula provides bounds on tinit\_complete based on dfi\_frequency encoding.

**Table 6-7 Max tinit\_complete Latency for LPDDR4X**

(dfi_freq_change)	Data Rate	dfi_ratio	userInputBasic.DramDataWidth	tinit_start+Tmrw (nDfiClks)+margins	tinit_start (nDfiClks)+margins	tinit_complete (nDfiClks)+margins
Cold Boot/Retention Exit	4264	4	max			6522
Relock only	4264	4	max	848	564	1970
Retention Entry	4264	4	max		116	716
Retrain + Relock	4264	4	max	848	564	4924
Retrain only	4264	4	max		158	2736
<hr/>						
Cold Boot/Retention Exit	4264	2	max			9440
Relock only	4264	2	max	1236	756	2932
Retention Entry	4264	2	max		130	1158
Retrain + Relock	4264	2	max	1236	756	6180
Retrain only	4264	2	max		158	3238

(dfi_freq_change)	Data Rate	dfi_ratio	userInputBasic.DramDataWidth	tinit_start+Tmrw (nDfiClks)+margins	tinit_start (nDfiClks)+margins	tinit_complete (nDfiClks)+margins
<hr/>						
Cold Boot/Retention Exit	3200	4	max			5602
Relock only	3200	4	max	736	504	1680
Retention Entry	3200	4	max		130	452
Retrain + Relock	3200	4	max	736	504	4064
Retrain Only	3200	4	max		158	2678
<hr/>						
Cold Boot/Retention Exit	3200	2	max			7440
Relock only	3200	2	max	1044	640	2296
Retention Entry	3200	2	max		130	880
Retrain + Relock	3200	2	max	1044	640	5046
Retrain only	3200	2	max		158	3024
<hr/>						
Cold Boot/Retention Exit	1600	2	max			6216
Relock only	1600	2	max	758	504	1502
Retention Entry	1600	2	max		130	452
Retrain + Relock	1600	2	max	758	504	4054
Retrain only	1600	2	max		158	2862
<hr/>						
Cold Boot/Retention Exit	662	2	max			2956
Relock only	662	2	max	626	402	724
Retention Entry	662	2	max		130	234
Retrain + Relock	662	2	max	626	402	746

Notes:

- DqsOscRunTimeSel=3 => 2048tCK; PPT enabled for 2 ranks
- The numbers in this table are valid only when PclkDCA is disabled. PclkDCA enabled adds the following:
  - 550 DfiClks

- Tmrw is the additional time for PIE to issue MRW instructions when DisableFspOp=0.
- These parameters are subject to change based on the final PIE FW.
- Data in above table is based on the following PHY configuration: lp5x4xcs2dq18ch2
- Values provided are maximum; actual times may be less

**Table 6-8 Max tinit\_complete Latency for LPDDR5X**

(dfi_freq_change)	Data Rate	dfi_ratio	userInputBasic.Nu mRanks	userInpu tBasic.Dr amData Width	userInpu tAdvanc ed.Disab leFSp	tinit_st art+Tm rw (nDfiCl ks)	tinit_start + 5% margin (nDfiClks)	tinit_comple te + 5% margin (nDfiClks)
Cold Boot/Retention Exit	9600	4	2	max	0			10322
Retrain + Relock	9600	4	2	max	0	2378	1750	7466
Retrain only	9600	4	2	max	0		176	3454
Relock only	9600	4	2	max	0	2378	1750	4226
Retention Entry	9600	4	2	max	0		110	1294
LP2 Enter/Exit	9600	4	2	max	0		294	2790
<hr/>								
Cold Boot/Retention Exit*	8533	4	2	max	0			9632
Retrain + Relock*	8533	4	2	max	0	2090	1624	7082
Retrain only	8533	4	2	max	0		176	3402
Relock only*	8533	4	2	max	0	2090	1612	3376
Retention Entry	8533	4	2	max	0		110	1160
LP2 Enter/Exit	8533	4	2	max	0		294	2096
<hr/>								
Cold Boot/Retention Exit*	7500	4	2	max	0			8960
Retrain + Relock*	7500	4	2	max	0	1966	1478	6710
Retrain only	7500	4	2	max	0		176	3352
Relock only*	7500	4	2	max	0	1966	1478	3098
Retention Entry	7500	4	2	max	0		110	1026
LP2 Enter/Exit	7500	4	2	max	0		294	1966
<hr/>								

<b>(dfi_freq_change)</b>	<b>Data Rate</b>	<b>dfi_ratio</b>	<b>userInputBasic.Nu mRanks</b>	<b>userInpu tBasic.Dr amData Width</b>	<b>userInpu tAdvanc ed.Disab leFSp</b>	<b>tinit_st art+Tm rw (nDfiCl ks)</b>	<b>tinit_start + 5% margin (nDfiClks)</b>	<b>tinit_comple te + 5% margin (nDfiClks)</b>
Cold Boot/Retention Exit*	6400	4	2	max	0			8222
Retrain + Relock*	6400	4	2	max	0	1728	1310	6262
Retrain only	6400	4	2	max	0		176	3286
Relock only*	6400	4	2	max	0	1728	1310	2786
Retention Entry	6400	4	2	max	0		110	874
LP2 Enter/Exit	6400	4	2	max	0		294	1822
Fast Relock + Retrain*	6400	4	2	max	0		1270	4962
<hr/>								
Cold Boot/Retention Exit*	5472	4	2	max	0			7626
Retrain + Relock*	5472	4	2	max	0	1632	1212	5948
Retrain only	5472	4	2	max	0		176	3252
Relock only*	5472	4	2	max	0	1632	1212	2590
Retention Entry	5472	4	2	max	0		110	756
LP2 Enter/Exit	5472	4	2	max	0		294	1706
Fast Relock + Retrain*	5472	4	2	max	0		294	4144
<hr/>								
Cold Boot/Retention Exit*	4200	4	2	max	0			6786
Retrain + Relock*	4200	4	2	max	0	1382	1032	5468
Retrain only	4200	4	2	max	0		176	3184
Relock only*	4200	4	2	max	0	1382	1032	2240
Retention Entry	4200	4	2	max	0		110	588
LP2 Enter/Exit	4200	4	2	max	0	1382	294	1554
Fast Relock + Retrain*	4200	4	2	max	0		294	4060
<hr/>								

<b>(dfi_freq_change)</b>	<b>Data Rate</b>	<b>dfi_ratio</b>	<b>userInputBasic.Nu mRanks</b>	<b>userInpu tBasic.Dr amData Width</b>	<b>userInpu tAdvanc ed.Disab leFSp</b>	<b>tinit_st art+Tm rw (nDfiCl ks)</b>	<b>tinit_start + 5% margin (nDfiClks)</b>	<b>tinit_comple te + 5% margin (nDfiClks)</b>
Cold Boot/Retention Exit*	3608	4	2	max	0			6408
Retrain + Relock*	3608	4	2	max	0	1306	956	5258
Retrain only	3608	4	2	max	0		176	3150
Relock only*	3608	4	2	max	0	1306	956	2086
Retention Entry	3608	4	2	max	0		110	514
LP2 Enter/Exit	3608	4	2	max	0		294	1474
Fast Relock + Retrain*	3608	4	2	max	0		294	4022
<hr/>								
Cold Boot/Retention Exit*	3200	2	2	x16	0			8126
Retrain + Relock*	3200	2	2	x16	0	1716	1296	6168
Retrain only	3200	2	2	x16	0		176	3376
Relock only*	3200	2	2	x16	0	1716	1296	3420
Fast Relock + Retrain*	3200	2	2	x16	0		1248	4852
Retention Entry	3200	2	2	X16	0		110	874
LP2 Enter/Exit	3200	2	2	X16	0		294	1622
<hr/>								
Cold Boot/Retention Exit*	2752	2	2	x16	0			7522
Retrain + Relock*	2752	2	2	x16	0	1632	1212	5852
Retrain only	2752	2	2	x16	0		176	3342
Relock only*	2752	2	2	x16	0	1632	1212	2388
Fast Relock + Retrain*	2752	2	2	x16	0		1180	4890
Retention Entry	2752	2	2	X16	0		110	756
LP2 Enter/Exit	2752	2	2	X16	0		294	1522
<hr/>								

<b>(dfi_freq_change)</b>	<b>Data Rate</b>	<b>dfi_ratio</b>	<b>userInputBasic.Nu mRanks</b>	<b>userInpu tBasic.Dr amData Width</b>	<b>userInpu tAdvanc ed.Disab leFSp</b>	<b>tinit_st art+Tm rw (nDfiCl ks)</b>	<b>tinit_start + 5% margin (nDfiClks)</b>	<b>tinit_comple te + 5% margin (nDfiClks)</b>
Cold Boot/Retention Exit*	2136	2	2	x16	0			6740
Retrain + Relock*	2136	2	2	x16	0	1374	1024	5410
Retrain only	2136	2	2	x16	0		176	3274
Relock only*	2136	2	2	x16	0	1374	1036	2048
Fast Relock + Retrain*	2136	2	2	x16	0		1026	4676
Retention Entry	2136	2	2	X16	0		110	598
LP2 Enter/Exit	2136	2	2	X16	0		294	1374
<hr/>								
Cold Boot/Retention Exit*	1672	2	2	x16	0			6136
Retrain + Relock*	1672	2	2	x16	0	1268	918	5082
Retrain only	1672	2	2	x16	0		176	3224
Relock only*	1672	2	2	x16	0	1268	900	1824
Fast Relock + Retrain*	1672	2	2	x16	0		846	4514
Retention Entry	1672	2	2	X16	0		110	480
LP2 Enter/Exit	1672	2	2	X16	0		294	1236
<hr/>								
Cold Boot/Retention Exit*	1600	2	2	x16	0			6026
Retrain + Relock*	1600	2	2	x16	0	1206	868	4980
Retrain only	1600	2	2	x16	0		176	3208
Relock only*	1600	2	2	x16	0	1206	856	2140
Fast Relock + Retrain*	1600	2	2	x16	0		828	4376
Retention Entry	1600	2	2	X16	0		110	454
LP2 Enter/Exit	1600	2	2	X16	0		294	1202
<hr/>								

(dfi_freq_change)	Data Rate	dfi_ratio	userInput.Basic.Nu mRanks	userInpu tBasic.Dr amData Width	userInpu tAdvanc ed.Disab leFSp	tinit_st art+Tm rw (nDfiCl ks)	tinit_start + 5% margin (nDfiClks)	tinit_comple te + 5% margin (nDfiClks)
Cold Boot/Retention Exit*	836	2	2	x16	0			2112
Retrain + Relock*	836	2	2	x16	0	1028	680	1592
Relock only*	836	2	2	x16	0	1028	662	1344
Retention Entry	836	2	2	X16	0		110	314
LP2 Enter/Exit	836	2	2	X16	0		294	1016
<hr/>								
Cold Boot/Retention Exit*	640	2	2	x16	0			1560
Retrain + Relock*	640	2	2	x16	0	928	578	1100
Relock only*	640	2	2	x16	0	928	578	1092
Retention Entry	640	2	2	X16	0		110	202
LP2 Enter/Exit	640	2	2	X16	0		276	832

Notes:

- The numbers in this table are valid only when NZQ=4.
- The numbers in this table are valid only when PclkDCA is disabled. PclkDCA enabled adds the following:
  - 550 DfiClks
- When using Deep Sleep Mode, tinit\_complete will increase by 190 us to account for LPDDR5 tXDSM\_XP
- DRAM ZQ calibration in Byte mode devices with multiple ranks must be serialized. The number specified is for worst case LPDDR5 Byte Mode devices. If x16 devices are used, ~1us can be saved from quoted number.
- Data in above table is based on the following PHY configuration: lp5x4xcs2dq18ch2.
- Values provided are maximum; actual times may be less.
- These parameters are subject to change based on the final PIE FW.



# 7

## Architecture

---

This chapter contains the following sections:

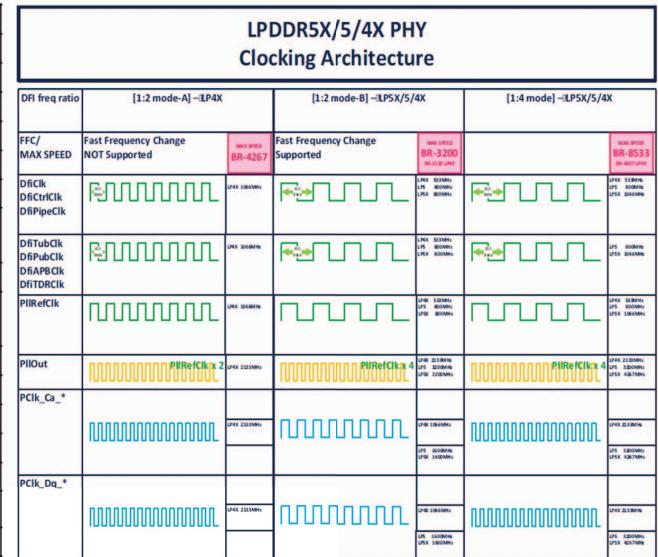
- “Clocking Architecture and Configurations” on page 186
- “PIPE Architecture Specification” on page 189
- “Configuration Ports” on page 193
- “Training Support Hardware Overview” on page 199
- “Interrupt Implementation” on page 206
- “Drift Tracking by MRR Snooping” on page 207
- “ARC-HS Microcontroller” on page 211
- “PHY PCLK DCA” on page 218
- “LPDDR5X/5 Specific Features” on page 220
- “LPDDR4X Specific Features” on page 225
- “Impedance Calibration” on page 227
- “Low Power Design” on page 234
- “Support of Dynamic Voltage Scaling” on page 235
- “Periodic Phase Training 2 (PPT2)” on page 237
- “Incremental Retraining Used in PPT2” on page 239

## 7.1 Clocking Architecture and Configurations

**Figure 7-1 Standard Product System Configuration**

### Clocking :: LPDDR5X/5/4X System Configuration

		LPDDR4X			LPDDR5X/5 *		Units
		2 b	2 b	4	2	4	
DFI	DFI freq ratio	2 b	2 b	4	2	4	1:2 or 1:4
	DfIClk freq	1066	533	533	800	1066	MHz
PLL	Pllout ratio	2	4	4	4	4	2x or 4x
	pllin	1066	533	533	800	1066	MHz
	pllout	2133	2133	2133	3200	4267	MHz
Clock Divider CK and CA	Clock divisor CA	1	2	1	2	1 / 2 <sup>c</sup>	1/1 or 1/2
	PclkCa	2133	1066	2133	1600	4267 / 2133	MHz
Clock Divider DQ/DQS	Clock divisor DQ	1	2	1	2	1	1/1 or 1/2
	PclkDq	2133	1066	2133	1600	4267	MHz
DRAM interface rates	CK bit rate	4267	2133	4267	1600	2133	Mbps
	CK frequency	2133	1066	2133	800	1066	MHz
	CA bit rate	2133	1066	2133	1600	2133	Mbps
	DQS bit rate	4267	2133	4267	3200	8533	Mbps
	DQS frequency	2133	1066	2133	1600	4267	MHz
	DQ bit rate	4267	2133	4267	3200	8533	Mbps



Notes:

a) LPDDR5X/5 JEDEC spec support 1:2 mode up to 3200 for backward compatibility

b) LPDDR4X first 1:2 mode in above table is mode A and 2nd 1:2 mode is mode -B.

c) In LPDDR5X/5 1:4 mode,

- when data rate  $\geq$  5500Mbps and pUserInputAdvanced->AcQuarterDataRate = 1, PclkDivCa\* = 2 is selected
- when data rate  $<$  5500Mbps and pUserInputAdvanced->AcQuarterDataRate = 1, PclkDivCa\* = 1 is selected
- when pUserInputAdvanced->AcQuarterDataRate = 0, PclkDivCa\* = 1 is selected.

**Figure 7-2 Standard Product System Configuration: PLL bypass**

## Clocking :: LPDDR5X/5/4X System Configuration : PLL bypass

		LPDDR4X			LPDDR5X/5		
	DFI freq ratio	2	2	4	2	4	Units
DFI	DfiClk freq	800	800	400	800	400	MHz
PLL	Pllout ratio	NA	NA	NA	NA	NA	2x or 4x
	pllin	1600	1600	1600	1600	1600	MHz
	pllout	1600	1600	1600	1600	1600	MHz
Clock Divider CK and CA	Clock divisor CA	1	1	1	1	1	1/1 or 1/2
	PclkCa	1600	1600	1600	1600	1600	MHz
Clock Divider DQ/DQS	Clock divisor DQ	1	1	1	1	1	1/1 or 1/2
	PclkDq	1600	1600	1600	1600	1600	MHz
DRAM interface rates	CK bit rate	3200	3200	3200	1600	800	Mbps
	CK frequency	1600	1600	1600	800	400	MHz
	CA bit rate	1600	1600	1600	1600	800	Mbps
	DQS bit rate	3200	3200	3200	3200	3200	Mbps
	DQS frequency	1600	1600	1600	1600	1600	MHz
	DQ bit rate	3200	3200	3200	3200	3200	Mbps

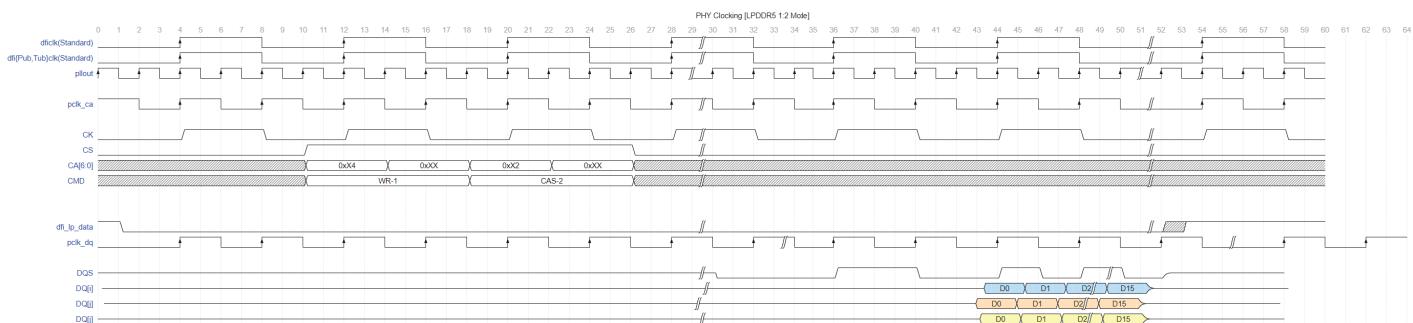
In PLL bypass mode, PllBypClk is half of data rate in MHz, and Clock divisor DQ and CA has to be 1 for all cases  
Fast frequency change is not supported when PLL is bypassed

Note: Fast frequency change is not supported when PLL is bypassed

**Figure 7-3 LPDDR5/5X-3200 1:2 Waveform**

## LPDDR5X/5-3200 1:2 Waveform

Supported range: 667 – 3200 Mbps in PLL mission mode; 50 – 3200 Mbps in PLL bypass



Max DRAM bit rates:

- CK: 1600 Mbps (800 MHz)
- CA: 1600 Mbps
- DQS: 3200 Mbps (1600 MHz)
- DQ: 3200 Mbps

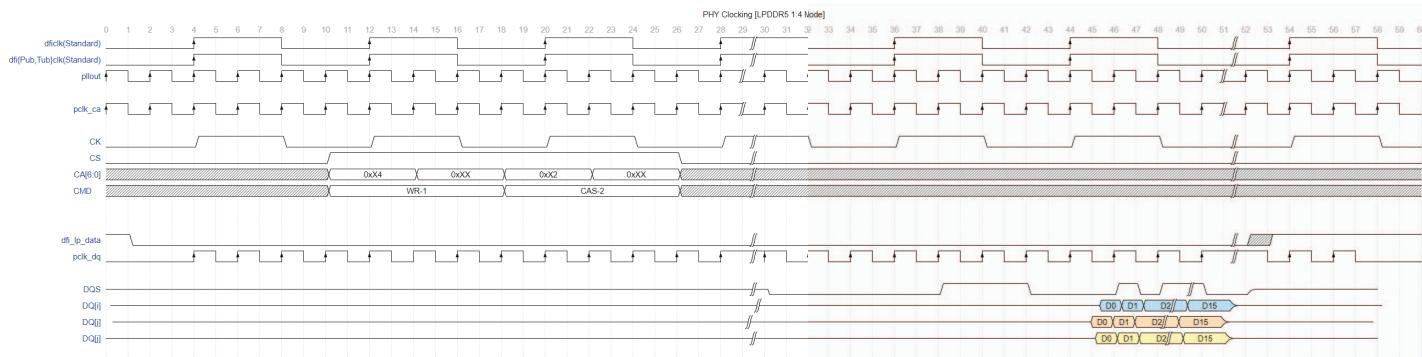
Max PLL clock rates:

- pllIn (DfiClk): 800MHz Standard Product (PMU: 800 MHz)
- pllout: 3200 MHz
- Pclk\_ca: 1600 MHz (=pllout / 2)
- Pclk\_dq: 1600 MHz (=pllout / 2)

**Figure 7-4 LPDDR5-6400 1:4 Waveform**

## LPDDR5-6400 1:4 Waveform

Supported range: 667 – 6400 Mbps in PLL mission mode; 50 – 3733 Mbps in PLL bypass



Max DRAM bit rates:

- CK: 1600 Mbps (800 MHz)
- CA: 1600 Mbps
- DQS: 6400 Mbps (3200 MHz)
- DQ: 6400 Mbps

Max PLL clock rates:

- pllin (DfiClk): 800MHz Standard Product (PMU: 800 MHz)
- pllout: 3200 MHz
- PCIk\_ca: 3200 MHz (=pllout / 1)
- PCIk\_dq: 3200 MHz (=pllout / 1)



When operating in LPDDR5X/5 mode, the AC can run at 1/4 pclk rate instead of the normal 1/2 rate pclk. When this is used, the AC LCDLs need 4UI range to not have a margin impact. Given this, this can only be used at or above 5500Mbps. This is a power savings feature, and can be enabled in phyinit.

## 7.2 PIPE Architecture Specification

### 7.2.1 Compile-time PIPE Overview

Pipeline staging flip-flops can be placed at the front end of the DFI interface, between the memory controller and PHY\_TOP to close synthesis timing issues due to customer specific layout/placement such as:

- Controller DFI pins are placed far away from PHY DFI pins OR
- Controller must need 50% DfiClk for input/output delays

To do so, set the following defines to a value of 1 if the pipeline stage is desired. If no pipeline stages are needed, leave these values undefined.

- DWC\_LPDDR5XPHY\_PIPE\_DFI\_WR
- DWC\_LPDDR5XPHY\_PIPE\_DFI\_RD
- DWC\_LPDDR5XPHY\_PIPE\_DFI\_MISC

PHY only supports following two options:

1. All DWC\_LPDDR5XPHY\_PIPE\_DFI\_MISC, DWC\_LPDDR5XPHY\_PIPE\_DFI\_WR, DWC\_LPDDR5XPHY\_PIPE\_DFI\_RD undefined, or
2. All DWC\_LPDDR5XPHY\_PIPE\_DFI\_MISC, DWC\_LPDDR5XPHY\_PIPE\_DFI\_WR, DWC\_LPDDR5XPHY\_PIPE\_DFI\_RD with values of 1

Conditional Macro	DFI input/output signals
DWC_LPDDR5XPHY_PIPE_DFI_MISC	dfi_reset_n dfi*_ctrlupd_ack dfi*_ctrlupd_req dfi*_phyupd_ack dfi*_phyupd_req dfi*_phyupd_type dfi*_dram_clk_disable dfi*_freq dfi*_freq_ratio dfi*_init_complete dfi*_init_start dfi*_phymstr_ack dfi*_phymstr_cs_state dfi*_phymstr_req dfi*_phymstr_state_sel dfi*_phymstr_type dfi*_address dfi*_cke dfi*_cs dfi*_lp_ctrl_ack dfi*_lp_data_ack dfi*_lp_ctrl_req dfi*_lp_data_req dfi*_lp_ctrl_wakeup dfi*_lp_data_wakeup dfi*_error dfi*_error_info
DWC_LPDDR5XPHY_PIPE_DFI_WR	dfi*_wrdata dfi*_wrdata_cs dfi*_wrdata_en dfi*_wrdata_mask  dfi*_wck_cs dfi*_wck_en dfi*_wck_toggle dfi*_wck_write dfi*_wrdata_link_ecc

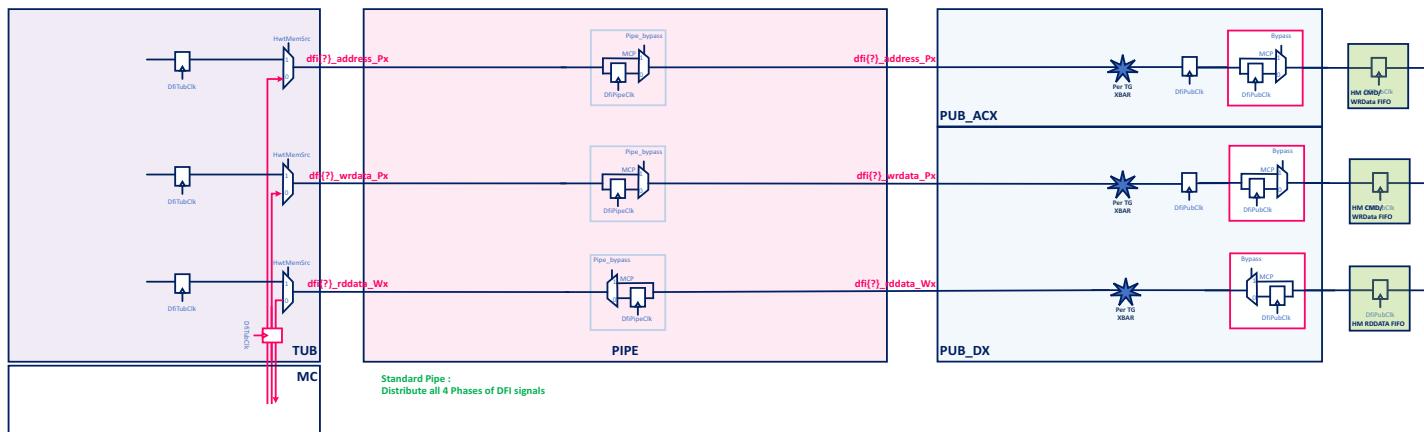
Conditional Macro	DFI input/output signals
DWC_LPDDR5XPHY_PIPE_DFI_RD	dwc_lpddr5xphy*_snoop_en dfi*_rddata_cs dfi*_rddata_en dfi*_rddata dfi*_rddata_valid dfi*_rddata_dbi

## 7.2.2 Configurable PIPE Overview

The PHY supports configurable PIPE modules to close physical synthesis timing between TUB and PUB blocks. These pipe modules are not used for any functional latency adjustment across command or data interfaces.

Basic “PIPE” module has a one pipeline flop with selectable bypass MUX.

**Figure 7-5 Configurable Pipeline**



**Table 7-1 Configurable Pipeline**

DfiClk (in MHz)	csrDxOutPipeEn[0] (in DfiClk)	csrDxInPipeEn[0] (in DfiClk)	csrAcInPipeEn[0] (in DfiClk)	csrDxWrPipeEn (in DfiClk)	csrDxRdPipeEn (in DfiClk)
> 800	1	1	1	0	0
<= 800	0	0	0	0	0

Note: Above pipeline settings apply independent of protocol (LPDDR4X or LPDDR5/5X) and DFI ratio (1:2 or 1:4).

Depending upon usage, multiple “PIPE” modules can be enabled as below:

1. DFI Address Input Pipeline:
  - a. Support one stage “PIPE” elements, selectable by csrAcInPipeEn[0]
  - b. Used by following Interface signals:
    - dfi\_reset\_n

- dfi\*\_address
- dfi\*\_dram\_clk\_disable
- dfi\*\_cs

## 2. DFI Data Interface Input Pipeline:

- a. Support 1 stage “PIPE” elements, selected by csrDxInPipeEn[0]
- b. Used by following Interface signals:

- dfi\*\_wrdata
- dfi\*\_wrdata\_cs
- dfi\*\_wrdata\_en
- dfi\*\_wrdata\_mask
- dfi\*\_wck\_cs
- dfi\*\_wck\_en
- dfi\*\_wck\_toggle
- dfi\*\_wck\_write
- dfi\*\_wrdata\_link\_ecc
- dwc\_lpddr5xphy\*\_snoop\_en
- dfi\*\_rddata\_cs
- dfi\*\_rddata\_en

## 3. DFI Data Interface Output Pipelines:

- a. Support 1 stage “PIPE” elements, Selected by csrDxOutPipeEn[0]
- b. Used by following interface signals

- dfi\*\_rddata\_db
- dfi\_rddata
- dfi\_rddata\_valid

## 7.3 Configuration Ports

The configuration port provides access to PUB registers. The configuration port includes an APB interface and a TDR serial interface. The configuration port interface resides in TUB.

### 7.3.1 APB

An APB interface can be selected to be the main access protocol for PUB configuration registers. The PUB APB interface is compatible to APB4. Refer to the AMBA APB Protocol Specification v2.0 for details about the APB protocol. The configuration port provides access to PUB registers.

The APB address (paddr[g-1:0]) points to a 32-bit data word. Therefore if the rest of the chip is using byte-based APB addresses, then the PUB APB address paddr\_arb[g-1:0] must be connected to paddr[g:2] at the chip level. Otherwise if 32-bit word addresses are used throughout the chip, then PUB paddr\_arb[g-1:0] must be connected to chip paddr\_arb[g-1:0].

The APB interface supports read register and write register accesses with wait states.

- If the APB Master adheres to the APB wait state protocol, there are no read register and write register issues.
- The PUB contains an asynchronous FIFO to synchronize APB requests from the APBCLK domain to the internal DFICLK domain.
  - The design is pipelined to enable posted writes at the maximum APB protocol rate of 1 write transaction/3 APBCLKs.
  - Read transactions causes the longest wait states, as these transactions cannot be posted and must complete before the PUB can accept more APB requests.
  - PHY generates even parity for APB read access, 1 bit per each 8 bit of read data.

#### 7.3.1.1 APB Interface Access Rules

The APB interface must be accessed under the following rules:

1. The APB host interface is 32 bit interface.
  - a. APB write data bits [31:16] from host should be always 0 when writing to internal PHY CSR.
  - b. APB read data bits [31:16] to host will be always 0 when reading from internal PHY CSR.
  - c. APB write to or read from external SRAM will always be in 32 bits.

2. In general, PHY does not support APB configuration. Registers should not be written or read during mission modes.
  - a. PHY does not support APB configuration registers access or SRAM access during mission modes except for lab debug purposes. The access from external APB Host and internal hardware sources (PMU and PIE) may collide and may result in unexpected behavior.
  - b. To support external APB access during mission mode debug, SOC must ensure following conditions:
    - i. Ensure memory controller stops sending following requests or completes any on-going such request.
      1. Dfi\_lp\_\*req
      2. Dfi\_init\_start
      3. Dfi\_ctrlupd\_req
      4. dwc\_lpddr5xphy\_snoop\_en
    - ii. Ensure memory controller stops sending following acknowledgments (or completes any on-going such request)
      1. Dfi\_phyupd\_ack
      2. Dfi\_phymstr\_ack
    - iii. Host programs PHY csrMicroContMuxSel=0
    - iv. Host programs PHY csrBlockDfiInterfaceEn=1
    - v. Host can write/read any PHY internal CSR or external SRAM locations.
    - vi. APB read is allowed while DFI memory write/read transactions are in progress.
    - vii. APB write is only allowed when DFI memory write/read transactions have completed and DFI interface is in idle state.



- Note**
- CSR writes may have functional impact.
  - Host must revert-back the CSR changes prior to re-starting DFI side band request.

3. The APB interface can only read/write mailboxes during training (while the PMU owns the CSR bus)
4. PSTRB functionality is not supported
5. The APB interface should be inactive, though not necessarily in reset, while the DfiClk is changing frequency.
6. PSLVERR not used (tied to 0).
7. No APB access allowed for at-least 10 DfiClk after de-assertion of PRESETn.
8. During APB read, each slave will generate even parity per 8 bits of read data and return it with RDATA to Host.

### 7.3.1.2 PPROT\_PIN Configuration and Purpose

- Typically tied off to a particular value, sometimes fuse programmed, or in some highly secure environments this is randomized by a different security engine.
- The PHY only responds to APB requests where PPROT\_APB matches the value of PPROT\_PIN.
  - This functionality prevents unauthorized accesses to the registers.

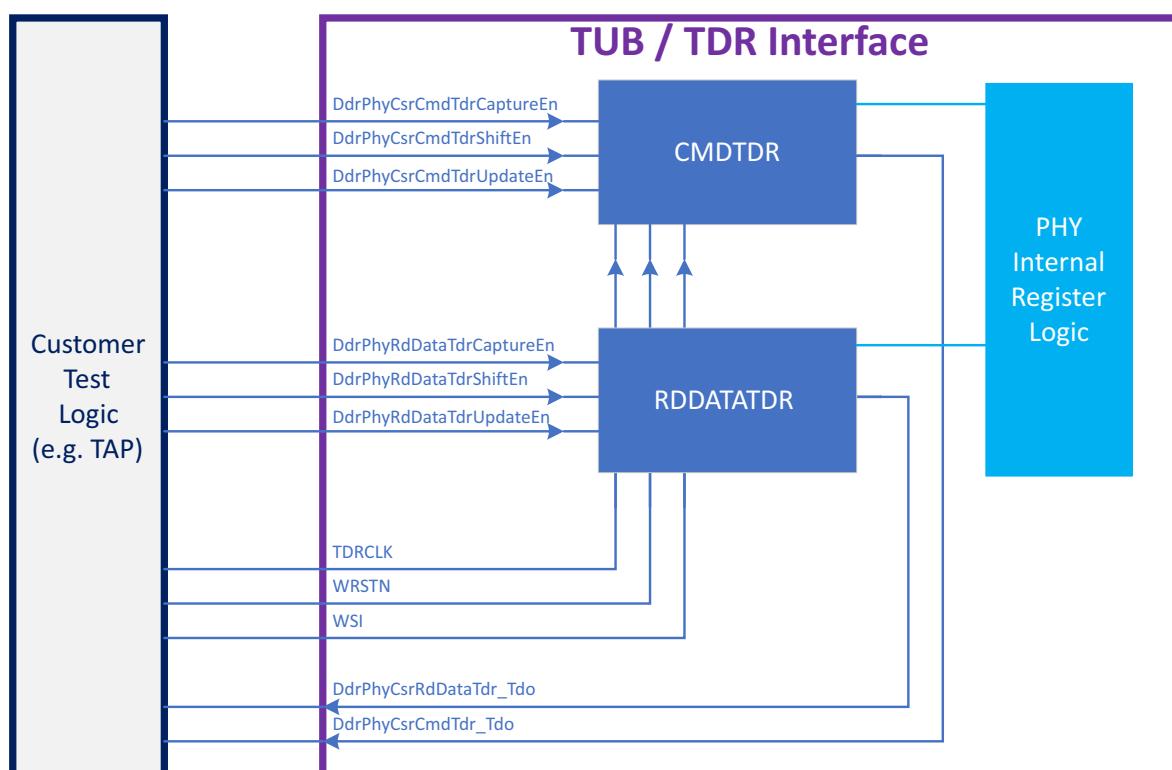
A hardware configurable port for comparison against the value on PPROT during activity PPROT\_PIN[2:0] is a static value provided by the SOC.

### 7.3.2 TDR

All of the PHY CSR address space can be accessed through a pair of TDR registers in the PHY. TDR (Test Data Register) and its design is based on the 1149.1 DR-Scan Sequence standard. For example, with a customer supplied TAP controller (optional), a JTAG port can serially access all of the PHY CSRs. This interface may be useful to access or trigger PHY test and/or debug modes without requiring full functional vectors. This method of accessing PHY CSRs can coexist with APB and microcontroller methods. The CSR access logic does not contain arbitration, so the TDR registers should be used only when APB and Microcontroller accesses are suspended. This condition happens naturally in most situations.

The following Figure shows an example implementing for JTAG access of CSRs from the SOC level.

**Figure 7-6 TDR interface**



### 7.3.2.1 CsrCmdTdr and CsrRdDataTdr Register Bit Definitions

**Table 7-2 CsrCmdTdr and CsrRdDataTdr Register Bit Definitions**

TDR Register Bit Names	Bit Definitions
CsrCmdTdr	
TDR_CsrCmdChain[60:29]	CSR Write Data, 32-bits
TDR_CsrCmdChain[28]	WR_RD=1 Write CSR WR_RD=0 Read CSR
TDR_CsrCmdChain[27:0]	CSR Address
CsrRdDataTdr	
TDR_RdDat[31:0]	CSR Read Data, 32-bits

### 7.3.2.2 TDR Procedure for CSR Write

The PHY TDR access procedure matches the access pattern for any user defined TDR. To Perform a CSR Write, a single transaction to TDR DdrPhyCsrCmdTdr is sufficient.

- Capture Phase [assert DdrPhyCsrCmdTdrCaptureEn for 1 clock cycle]
  - Assert DdrPhyCsrCmdTdrCaptureEn for 1 clock cycle
  - This TDR does not have defined capture data .
- Shift Phase [Assert DdrPhyCsrCmdTdrShiftEn for 61 clock cycles]
  - Serially apply 28-bits of CSR address, followed by the WR\_RD bit set to 1, followed by the 32-bits of CSR write data to the WSI input.
- Update Phase [Assert DdrPhyCsrCmdTdrUpdateEn for 1 clock cycle]
  - Applies the write command to the CSR path.

### 7.3.2.3 TDR Procedure for CSR Read

The PHY TDR access procedure matches the access pattern for any user defined TDR.

To Perform a CSR Read, two transactions are required. The first transaction is to TDR DdrPhyCsrCmdTdr to perform the internal command. The second transaction is to TDR DdrPhyCsrRdDataTdr to capture the data.

- First TDR Transaction to DdrPhyCsrCmdTdr
- Capture Phase [assert DdrPhyCsrCmdTdrCaptureEn for 1 clock cycle]
  - Assert DdrPhyCsrCmdTdrCaptureEn for 1 clock cycle
  - This TDR does not have defined capture data.
- Shift Phase [Assert DdrPhyCsrCmdTdrShiftEn for 61 clock cycles]
  - Serially apply 28-bits of CSR address, followed by the WR\_RD bit set to 0, followed by the 32-bits of zero to the WSI input.

- Update Phase [Assert DdrPhyCsrCmdTdrUpdateEn for 1 clock cycle]
  - Applies the read command to the CSR path.
- Wait at least 10 TDRCLK cycles
- Second TDR Transaction to DdrPhyCsrRdDataTdr
- Capture Phase [assert DdrPhyCsrRdDataTdrCaptureEn for 1 clock cycle]
  - Assert DdrPhyCsrCmdTdrCaptureEn for 1 clock cycle
  - The TDR captures the internal CSR Read data.
- Shift Phase [Assert DdrPhyCsrRdDataTdrShiftEn for 32 clock cycles]
  - Observe DdrPhyCsrRdDataTdr\_Tdo to see results of the Captured data. Incoming WSI may be any don't care value (zeros recommended).
- Update Phase [Assert DdrPhyCsrRdDataTdrUpdateEn for 1 clock cycle]
  - Has no effect for this TDR.



**Note** This step is not necessary in the design. However, it is included because the IEEE 1149.1 Specification for TDR accesses indicate that the update is required.

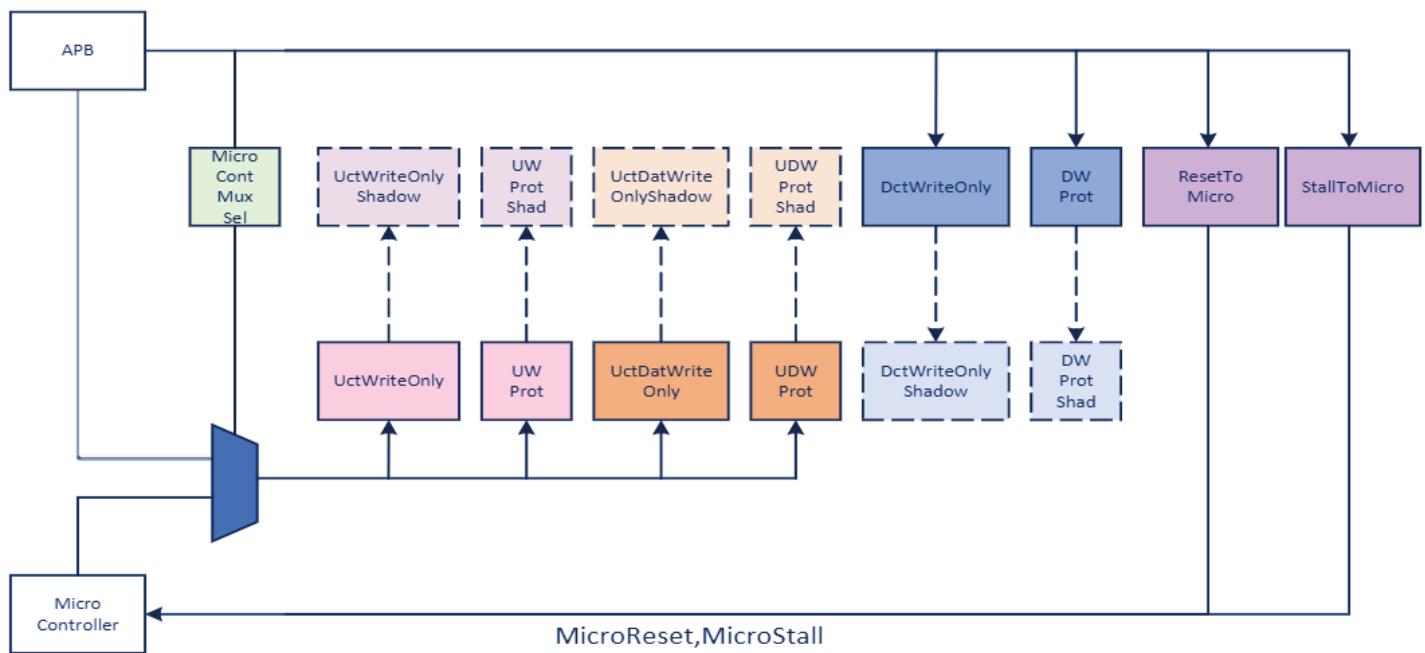
### 7.3.3 Training Sequence

The training firmware is loaded and controlled by accessing CSRs and memory.

At the end of training, the DRAMs will be left in the Self-Refresh state. The memory controller is expected to take the DRAMs out of the Self Refresh state and start mission mode traffic using the same PState that was last trained. However, because the training firmware puts the DRAMs into the Self-Refresh state using a SRE command, the DRAMs will use the same termination, VREF, and frequency settings they were using when they were taken out of the Self Refresh state.

### 7.3.4 MailBox

Since the PHY CSR bus is controllable by only one client at a time, the bus is generally unavailable to the APB port while training (when the PHY Micro-Controller (PMU) has control of the CSR bus). In order to enable successful message passing and handoff between the PMU and the higher level boot-code, a set of protocol CSRs (with active and shadow copies) are provided as below.

**Figure 7-7 Message passing CSRs**

### 7.3.4.1 MailBox Messaging Protocol

The protocol operates in the following manner:

- All protocol bits and internal state are initialized to zero
- The internal state bits follow the protocol bits
- The uCtl follows/mirrors BIOS protocol/state bits
- The uCtl knows a new command is ready when it sees the MC protocol != uCtl internal state.
- BIOS knows the uCtl has acted upon its command when it sees its internal state reflected on the uCtl's protocol bit: uCtl protocol == BIOS internal state
- BIOS drives the initial transaction.
- BIOS/PMU use the same handshake protocol if the UctDatWriteOnly mailbox is used.
  - Agreed context between uCtl and BIOS DctWriteOnly/UctWriteOnly messaging will signal which upstream mbox the uCtl will use. This is to maximize production code reuse.

## 7.4 Training Support Hardware Overview

For DDR training the TUB contains a microcontroller also known as the PMU (PHY Micro-control Unit) and, within the PHY, Hardware Training (HWT) engines that perform the low level training work. The PMU is responsible for responding to and communicating with external control agents such as System Software, orchestrating the overall training flow, initializing and starting the HWT engines through CSR Bus accesses and performing the mathematical operations needed for discovering for the optimal data eye sampling position. All the HWT logic runs off of a gated version of DfiClk, which is referred to as HClk in the description that follows. There are three different types of HWT engines employed in training.

- An Address Command State Machine (ACSM) type that is responsible of generating DRAM commands to the memory interface and controlling the sequencing of the other two engines.
- The HWT engine is the PRBS Pattern Generator Checker (PPGC) responsible for generating data patterns used in training.
- The Data Training State Machine (DTSM) is responsible for implementing the detector and feedback control loop used to lock to the eye contour or find sampled transition regions. DTSM will be HWT engine that will be in the PUB\_DX.

It is important to note that the HWT hardware lives on the same side of the CMD (Command), TX (Transmit) & RX (Receive) FIFOs as the controller. There is a mux that sits at the interface of the PIPE that selects the DFI bus from MC or from HWT as the command source. The mux select line is controlled by the HwtMemSrc CSR bit. Access to the PHY is exclusive as only one source, HWT or controller, may drive the interface at a time.

System SW is responsible for loading the PMU local SRAM, providing DRAM configuration information through the message buffer, and then, instructing the PMU to start its cold boot training sequence. The PMU SRAM is accessible through the CSR interface. In general, system SW should not access PMU SRAM while the PMU is executing instructions. To reliably access SRAM, system SW must put the PMU back into reset or the stalled state.

The PMU causes DRAM traffic to flow through the PHY by programming the Address Command State Machine (ACSM). There is one ACSM block that serves 2 channel. DDR data streams are generated by the PRBS Pattern Generator Checker block (PPGC). There is one PPGC block and there is an independent TX and RX generator function.

The PRBS engine is fully programmable and can generate a data stream for any 16-bit or less polynomial. The PPGC can generate the same polynomial for each DQ-bit within the DBYTE. However, it also has the flexibility to be partitioned to drive different, but smaller, polynomials into arbitrary groups of DQ bits. This flexibility is achieved partly through the architecture of the PRBS state block and partly through the architecture of the programmable tap selection block. The PPGC supports any PRBS state block tap selection and any PRBS stride (advance) rate on the 16-bit PRBS logic making it extremely flexible for generating arbitrary and random training sequences. A separate generator and checker block is provided to simplify programmed check synchronization of complex DDR read and write streams. The PPGC provides a fixed PRBS31 pattern. It also supports deterministic pattern generation in addition to PRBS generation. There is a 32-bit pattern register per DQ bit in a DBYTE. The pattern register shares the same registers as used in the tap selection function. Therefore, if switching between PRBS and pattern mode, the PMU must save the CSR state to SRAM if maintaining state across pattern mode invocations is important.

The Data Training State Machine block (DTSM) is responsible for comparing the expected data sequence against the sequence that was sent and returned through the PHY interface. This includes data streams that are looped back on the Pclk\_Ca/Pclk\_Dq side of the PHY TX/RX interface and sequences that are read and written from DRAM. DTSM can control per lane within DBYTE. Prior to starting a training loop, the PMU engages the DTSM to control a TX/RX, Voltage/Delay training knob through a CSR. Once engaged, the

PMU then programs the ACSM and PRBS blocks to drive the desired training sequence. When training has begun, the DTSM compares the two streams it receives and determines whether to increment or decrement the control knob. The DTSM implements data filters and a binary adder with upper and lower threshold comparison logic to train to a data contour eye position. Each DTSM operates independently of the other DTSMs, but the time spent in the training loop is globally determined by the ACSM sequence.

#### 7.4.1 SRAM

	LPDDR5X SRAMs			
	Without ECC/Parity		With ECC/Parity	
	Size	Width	Size	Width
ICCM0	1 x 48KB	64 bits	1 x 58.5KB	78 bits
ICCM1	1 x 48KB	64 bits	1 x 58.5KB	78 bits
DCCM0	1 x 24KB	32 bits	1 x 29.25KB	39 bits
DCCM1	1 x 24KB	32 bits	1 x 29.25KB	39 bits
DCCM2	1 x 24KB	32 bits	1 x 29.25KB	39 bits
DCCM3	1 x 24KB	32 bits	1 x 29.25KB	39 bits
BC0	1 x 2KB	64 bits	No ECC/Parity	No ECC/Parity
BC1	1 x 2KB	64 bits	No ECC/Parity	No ECC/Parity
PT0	1 x 1KB	8 bits	No ECC/Parity	No ECC/Parity
PT1	1 x 1KB	8 bits	No ECC/Parity	No ECC/Parity
ACSM	1 x 16KB	64 bits	1x 18KB	72 bits
PIE	1x 12.5KB	50 bits	1 x 14KB	56 bits

- By default, the PHY expects that the ICCM and DCCM SRAMs are wide enough to support ECC data, and an interrupt is generated (if enabled) when an ECC error is detected. If the ICCM and DCCM SRAMs are not wide enough to support ECC data, ECC must be disabled using `csrArcPmuEccCtl`.
- BC (Branch Cache) and PT (Prediction Table) SRAMs do not support ECC or parity. A data error in these memories only causes mis-prediction, it affects the performance but not the functionality. These SRAMs are not accessible to Host by APB bus.
- ACSM and PIE SRAM support byte-level even parity; an interrupt will be generated when parity error is detected.
- PHY supports parity poisoning logic to force parity error (used to check the integrity of parity checker).
- PHY supports Zero-fill for DCCM SRAM

**Table 7-3 PIE SRAM Address Map**

<b>22:20 (ppp)</b>	<b>19:16 (ttt)</b>	<b>15:12 (cccc)</b>	<b>11:1</b>	<b>0</b>
000	0100	0100	Selects 50 bit data row	0: data bits 31:0 1: data bits 49:32

**Table 7-4 ACSM SRAM Address Map**

<b>22:20 (ppp)</b>	<b>19:16 (ttt)</b>	<b>15:12 (cccc)</b>	<b>11:1</b>	<b>0</b>
000	0100	0001	Selects 64 bit data row	0: data bits 31:0 1: data bits 63:32

In some applications, SOC may put external SRAM in sleep mode (reduced power without losing content). PHY does not provide any logic to control external SRAMs power gating.

<b>SRAM</b>	<b>When SRAMs can be powered down by SOC?</b>	<b>When SRAMs must be powered up?</b>	<b>Note</b>
ICCM{0,1}	Mission mode ( dfi_init_complete=1)	During Initialization or training, that is, dfi_init_complete=0.	
DCCM{0,1,2,3}	Mission mode ( dfi_init_complete=1)	During Initialization or training, that is, dfi_init_complete=0.	
BC{0,1}	Mission mode ( dfi_init_complete=1)	During Initialization or training, that is, dfi_init_complete=0.	
PT{0,1}	Mission mode ( dfi_init_complete=1)	During Initialization or training, that is, dfi_init_complete=0.	
ACSM	Mission mode ( dfi_init_complete=1) AND NOT in any MC initiated Frequency Change Request state ( Host/UMC should have some Flag to indicate all such states)	1. During Initialization or training, that is, dfi_init_complete=0. 2. During any MC initiated Frequency Change state.	
PIE	Mission mode ( dfi_init_complete=1) AND NOT in any MC initiated Frequency Change Request state ( Host/UMC should have some Flag to indicate all such states)	1. During Initialization or training, that is, dfi_init_complete=0. 2. During any MC initiated Frequency Change state.	

## 7.4.2 PIE

The PHY Initialization Engine (or PIE for short) is the hardware module that is designed to perform these MC interactions. The PIE is used in the PHY to facilitate MC interactions on the following features and events:

- DFI Status Interface
  - Frequency changes

- Issue PHY re-training without frequency change PPT
- Entering/Exiting LP3 IO retention
- Enter/Exit LP2
- DFI PHY Master Interface (PMI)

The MC initiates these handshake requests based on DFI spec. This document describes the overall hardware blocks, their SW programing, and their interactions to better understand PHY internal operation on these requests.

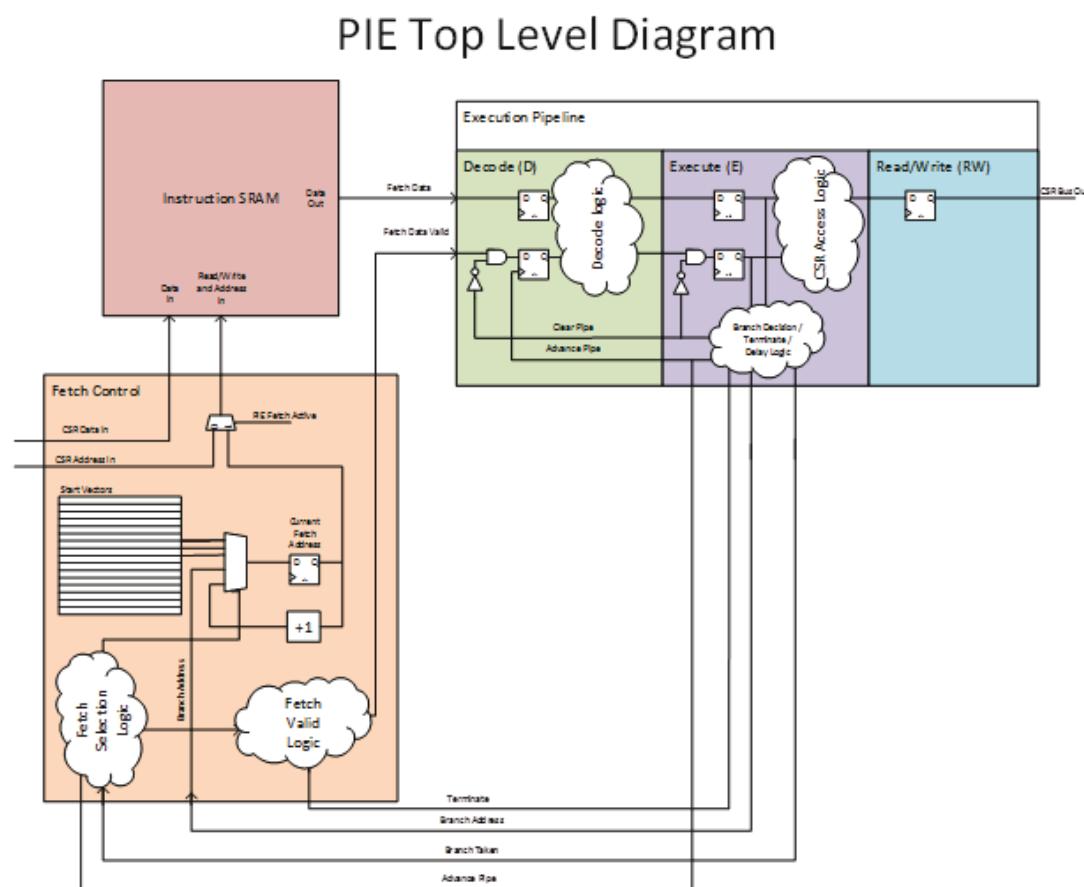
The PIE is a programmable micro sequencer that writes PHY registers (CSR's). The PIE is used to transition the PHY from one state to another, based on different programmable register write sequences. It also can introduce programmable delay between register writes within the sequence. The microcode program that the PIE runs is loaded into SRAM during the initialization process performed as part of the PhyInit sequence. This programmability allows sequences to change if required post-silicon tapeout or production. For more information on PhyInit process and configurability, see the PhyInit documentation.

#### 7.4.2.1 PIE High Level Overview

The overall sequencer architecture is shown in the diagram below. The PIE operates as a simple fetch/execute machine. It can execute up to 1 micro-instruction per DfiClk cycle, but instructions can be programmed to take longer than a DfiClk cycle by using the delay field of each instruction.

The PIE is divided into three major sections:

- Instruction Memory – Stores the microcode program that will be executed
- Fetch Control – Controls the instruction fetch operations and informs the pipeline when it has valid instructions to execute.
- Execution Pipeline – Captures, decodes, and executes the instructions. Also interfaces to the CSR bus to perform the writes and reads.

**Figure 7-8 PIE Top Level Diagram**

The Fetch Control (FC) unit is the block that does the following:

- Determines when the PIE is running/fetching
- Determines where the PIE starts executing
- Determines when the SRAM is being accessed and by which source
- Determines what address/controls/data are going to the SRAM
- Controls the valid bit to the Execution Pipeline

The Execution Pipeline unit is the block that does the following:

- Captures the fetched instructions from the SRAM
- Decodes the instructions to understand what to do
- Performs the operation of the instruction
- Interfaces with the CSR bus to perform writes/reads and creates the address/data for these accesses
- Creates the pipeline advance, branch, terminate and clear pipe signals that are sent to the Fetch Control unit
- Manages the GPRs for WRITE\_GPR/READ\_GPR/TEST instructions
- Manages the Zero (Z) flag used by the branch instructions

### 7.4.2.2 PIE Instruction SRAM

The PIE microcode engine uses an SRAM external to the PHY to store the microcode program that is executed when the PIE is activated. The SRAM will have the following features:

- Single ported memory (single read/write port)
- 56 bits wide for data (includes parity)
- 2048 entries deep. The pie\_ram\_addr[11] bit is RFU and can be left dangling.
- 1 cycle access (Data is available on the output 1 DFI clock cycle after the address is presented)
- Access to the SRAM will be through the APB or Config bus
  - When the PIE is not running, the contents of the SRAM can be written and read.
  - When the PIE is running, it will not be possible to access the SRAM contents.
    - Writes will be ignored
    - Reads will return zeros
  - In other words, the control/address/data into the SRAM should be a simple mux with the select controlled by whether the PIE is running or not.
  - ARC has access to PIE SRAM.

The PIE will be designed so that the SRAM address for reads will come directly out of registers while the PIE is running. The SRAM read data will be immediately registered by the execution pipeline with no logic before the registers. This is done to ease timing as much as possible for the SRAM interfaces.

The SRAM memory is organized as (up to) 2048 entries, each 56 bits in width. 50 bits are for data, and 6 bits are for parity. Parity is calculated As per table below.

Parity Bit in SRAM	Associated Data Bits
55	49:41
54	40:32
53	31:24
52	23:16
51	15:8
50	7:0

Even parity is calculated by the hardware whenever the SRAM is written, and then stored in the SRAM along with the write data. Parity is checked by the hardware whenever the PIE reads the SRAM during PIE operation.

When enabled and not masked, Parity errors will assert interrupt output pin with a sticky status flag but operation will continue. Parity errors indicate that the PIE instructions are defective and will no longer function properly, and software should treat these errors as fatal.

In designs where SRAM data integrity is guaranteed and parity checking need not be implemented, a 50-bit RAM can be used. In this case, the parity bit inputs to the PHY pie\_data\_dout[55:50] must be tied low, and the parity interrupt bit should be disabled.

The PIE SRAM memory is mapped into the CSR address space. The parity bits are not accessible by software. The complete address for SRAM entry is shown in the following table:

**Table 7-5 PIE SRAM Address Map**

<b>22:20 (ppp)</b>	<b>19:16 (tttt)</b>	<b>15:12 (cccc)</b>	<b>11:1</b>	<b>0</b>
000	0100	0100	Selects the 50-bit data row	0: 32 LSB 1: 18 MSBs

Note that the SRAM memory is not PState-able.

## 7.5 Interrupt Implementation

The PHY has dedicated active low output signals – PhyInt\_n[15:0]. This output has five registers which are used to determine the state of the signal – PhyInterruptEnable, PhyInterruptMask, PhyInterruptClear, PhyInterruptFWControl and PhyInterruptStatus.

The Firmware posts interrupts to the PhyInterruptFWControl as appropriate. Once the PhyInterruptFWControl register is written the values will remain unchanged until the Firmware re-writes the register. There is an edge detector at the output of the PhyInterruptFWControl register which then feeds into the PhyInterruptStatus register, this way an interrupt will only be generated when the PhyInterruptFWControl register first writes a value from 0 to 1. Because of this, after posting an interrupt, the firmware must re-write the register bit to 0 before again setting it to a 1.

The PhyInterruptEnable register is used to either allow or dis-allow an interrupt's status to appear in the PhyInterruptStatus register. In order for the interrupt to be registered in this read only register the corresponding bit in the PhyInterruptEnable register must already be set to 1 at the time the edge detector fires. It is possible to see an interrupt in PhyInterruptFWControl but not in PhyInterruptStatus if the corresponding bit in PhyInterruptEnable is 0.

Once an interrupt is registered in PhyInterruptStatus it will be used to assert the corresponding bit of PhyInt\_n output signal if the corresponding bit in the PhyInterruptMask register is 0.

It is possible to clear the interrupts using the PhyInterruptClear register. Writing a 1 to any bit of this register clears the corresponding interrupt in the PhyInterruptStatus register. The PhyInterruptClear register is always read as 0.

When an interrupt is cleared from the PhyInterruptStatus register, the corresponding PhyInt\_n output signal will negate.

Clearing an interrupt from the PhyInterruptStatus register this does not affect the interrupt status in the PhyInterruptFWControl register. It is possible to clear an interrupt from the PhyInterruptStatus register but still see the corresponding interrupt indicated in the PhyInterruptFWControl register. However, since an edge detector is used that interrupt will not be re-asserted in the PhyInterruptStatus register until it is cleared to 0 in the PhyInterruptFWControl register and re-written to a 1.

In the event that an interrupt posts in the same clock cycle that is it cleared, the posting operation will take precedence.

Some bits of the csrPhyInterruptStatus register are duplicated as two adjacent anti-valent bits, and are output on primary PHY outputs PhyInt\_fault[5:0] as shown in [Table 7-6](#). A PhyInterruptOverride register is also provided to allow the user to emulate the existence of any interrupt condition. Refer to ["Register Descriptions"](#) on page [349](#), for more detail description of each register.

**Table 7-6 Anti-Valent Interrupts**

Interrupt	Active Hi Bit	Active Lo Bit
PhyAcsmParityErr	PhyInt_fault[0]	PhyInt_fault[1]
PhyPIEParityErr	PhyInt_fault[2]	PhyInt_fault[3]
PhyRdfPtrChkErr	PhyInt_fault[4]	PhyInt_fault[5]

## 7.6 Drift Tracking by MRR Snooping

SDRAMs have internal oscillators to measure the tWCK2DQI/tWCK2DQO for LPDDR5 timing parameters. PHY implements hardware to snoop the oscillator values during mission mode and detect any drift in write or read path due to voltage and temperature changes.

When enabling MRR Snoop retraining, PPT retraining should be executed during Relock+Retrain or Fast Relock+Retrain sequence when frequency changes.

PHYINIT UserInput settings must be as below:

- UserInputAdvanced->EnWck2DqoTracking = 1
- UserInputAdvanced->DisableRetraining = 0 (When SkipTrain = 1, DisableRetraining = 1 is mandatory. PHY DRAM drift compensation retraining can only be enabled if Training firmware is enabled)
- UserInputAdvanced->RetrainMode = 1
- UserInputAdvanced->RxClkTrackEn = 1
- UserInputAdvanced->PhyMstrTrainInterval = 0
- UserInputAdvanced->DisablePhyUpdate = 1



- The MRR snoop feature is not supported in LPDDR4X mode.
- When Snoop feature is enabled, following retraining options are not supported:
  - Retrain Only (dfi\_frequency = 0xc) using DFI Status Interface
  - Retraining using DFI PHY Master Interface

MRR Snoop sequence is

1. Controller sends a MPC command to start the oscillator.
2. DRAM MR37 and MR40 are already programmed to set oscillator runtime of 2048 cycles during retraining
3. PHY always uses TrackingModeCntrl\_pX.DqsOscRunTimeSel = 0x3 (default)
4. Reads the Oscillator Results
  - After oscillator runtime expires, controller can send MRR command to read oscillator values.
  - Controller should also assert dwc\_lpddr5xphy0\_snoop\_en.

Controller can take the liberty of flexible scheduling of these MPC, MRW and MRR commands during mission mode.

Using the dwc\_lpddr5xphy0\_snoop\_en mechanism, the Controller is solely responsible for ensuring the quality of the measurement that is read by MRRs. This means that once the DRAM oscillator is started, the Controller guarantees it completes without interruption of continuous and steady operation of the CK\_t/c clocks.

Once controller starts oscillator, it should not assert following DFI signals:

- Dfi\_ctrlupd\_req
- Dfi\_lp\_ctrl\_req
- Dfi\_dram\_clk\_disable
- Dfi\_init\_start (Frequency change)

Controller must send dfi\_ctrlupd\_req periodically (when oscillator is not running) to update the new delays and ZCal codes to IO, where PHY will stop the DRAM clock for CK IO code update. Before asserting dfi\_ctrlupd\_req with MRR Snoop enabled, DFI interface should be in IDLE state and DRAM should be in idle or refresh or active power-down state to meet clock stop requirements as defined in JEDEC.

Under multiple PState configuration, if the system uses MRR Snoop retraining in any PStates, PHYINIT Userinput EnWck2DqoTracking = 1 is required for all PStates.

For the PState using MRR Snoop retraining,

- Controller runs DRAM OSC /MRR Snoop for retraining.
- During dfi\_ctrlupd\_req/ack sequence, LCDL calibration, ZCal code update, and delay compensation will be executed.

For the PState not using MRR Snoop retraining,

- Controller does not run DRAM OSC /MRR Snoop for retraining.
- During dfi\_ctrlupd\_req/ack sequence, LCDL calibration and ZCal code update will be executed.



- Synopsys controller cannot stop dfi\_ctrlupd\_req while oscillator is running. Instead, controller will provide output dwc\_lpddr5xphy\_snoop\_osc\_running (per channel) which will be asserted (1'b1) before MPC command to start DRAM oscillator and de-asserted (1'b0) after expiring OSC runtime (2048tCK) from the last MPC(OSC start) command. PHY will use it to mask IO ZCal code update due to dfi\_ctrlupd\_req when dwc\_lpddr5xphy\_snoop\_osc\_running input is asserted. PHY will not use this input for any other function.
- If 3rd party controller can stop dfi\_ctrlupd\_req while oscillator is running, PHY input dwc\_lpddr5xphy\_snoop\_osc\_running should be tied LOW.

The controller asserts dwc\_lpddr5xphy0\_snoop\_en[3:0] flag, along with dfi\_rddata\_en/dfi\_rddata\_cs for full burst-length UIs, to enable the PHY to capture the MRR data.

<b>PHY Inputs</b>	<b>LPDDR5 mode</b>
dwc_lpddr5xphy0_snoop_en[3]	Asserted when MR39 is read by controller.
dwc_lpddr5xphy0_snoop_en[2]	Asserted when MR38 is read by controller.
dwc_lpddr5xphy0_snoop_en[1]	Asserted when MR36 is read by controller.
dwc_lpddr5xphy0_snoop_en[0]	Asserted when MR35 is read by controller.

The dwc\_lpddr5xphy0\_snoop\_en is 4 bit wide for each phase of dfi\_rddata\_en. So for 32 bit PHY, dwc\_lpddr5xphy0\_snoop\_en\_P{0..3} each will be  $4 \times 4 = 16$  bits wide.

In two ranks system, MC needs to complete a pair of MRR commands on one rank before sending next pair of MRR commands on 2nd rank.

PHY will calculate drift and new Delay values when it has received results for both MRR for a timing group. PHY will update Delay CSRs when next DFI Update event occurs.

The difference between MRR Snoop retraining and PPT retraining(Retrain Only/ PMI) is summarized in Table [Table 7-7](#).

**Table 7-7 Comparison Between PPT (Retrain Only/PMI) and MRR Snoop Retraining**

Item	PPT1 retraining	PPT2 retraining	DFI MRR SNOOP retraining (Controller assisted drift tracking by MRR)
Trigger mechanism	DFI Status Interface : dfi_init_start/complete with dfi_frequency = 0xc or DFI PHY Master Interface : dfi_phymstr_req/ack	Synopsys proprietary DFI Update Interface : dfi_ctrlupd_req/ack with dfi_ctrlupd_type = 2'b01.	Synopsys proprietary DFI Snoop Interface : dwc_ddrphy_snoop_en [3:0] and DFI Update Interface : dfi_ctrlupd_req/ack
PHYINIT UserInput	pUserInputAdvanced->DisableRetraining = 0 pUserInputAdvanced->RetrainMode = 1 pUserInputAdvanced->PhyMstrTrainInterval = Non zero pUserInputAdvanced->RxClkTrackEn = 1	pUserInputAdvanced->DisableRetraining = 0 pUserInputAdvanced->RetrainMode = 4 pUserInputAdvanced->RxClkTrackEn = 1	pUserInputAdvanced->EnWck2DqoTracking = 1 pUserInputAdvanced->DisableRetraining = 0 pUserInputAdvanced->RetrainMode = 1 pUserInputAdvanced->PhyMstrTrainInterval = 0 pUserInputAdvanced->RxClkTrackEn = 1 pUserInputAdvanced->DisablePhyUpdate = 1
Limitation	Retrain Only cannot be supported with MRR Snoop	This method can be supported with PPT1 This method cannot be supported with MRR Snoop	This method cannot be supported with Retrain Only. Not supported in LPDDR4X mode.
Retraining target	All (RxClk,RxEn and TxDQ)-retrainings for both channels and both ranks are all performed during tinit_complete period	Incremental (RxClk or RxEn or TxDQ) retraining in a single channel/rank is performed at a time.	RxClk, RxEn and TxDQ for both ranks in a single channel are updated at a time by DFI Update sequence
Retraining duration	Refer to tinit_complete value LP5-6400Mbps : tinit_start + tinit_complet = ~3600 DfiClks (4.5us)	500ns or lower at >= 3200Mbps for each iteration. For dual channel, 2 rank system with link ECC enabled, total 16 iterations are required.  This retraining time can be partially or completely hidden with DRAM refresh tRFCab timing.	MPC and MRR commands for oscillator are sent by MC without stalling the mission mode traffic.  Refer to tctrlupd_max value for traffic stall duration : 25 DfiClks

Item	PPT1 retraining	PPT2 retraining	DFI MRR SNOOP retraining (Controller assisted drift tracking by MRR)
VT-drift detection mechanism	TxDQ : based on DRAM oscillator value (WCK2DQI) RxEn : based on RDQS from DRAM (WCK2DQO) RxClk : based on RxReplica circuit	TxDQ, RxClk : PHY performs WR/RD-FIFO operation with simple data pattern to detect write/read eye drift.  RxEn : based on RDQS from DRAM  Higher accuracy as DRAM oscillators are not used. Write timing budget is improved with PPT2 owing to no DRAM OSC error.	TxDQ : based on DRAM oscillator value (WCK2DQI) RxEn : based on DRAM oscillator value (WCK2DQO) RxClk : based on RxReplica circuit  RxEn and TxDQ oscillators are started and read by controller during mission mode and delays are updated during DFI Update.
DRAM State	Self-refresh	Idle, Self-refresh, Refresh	Idle, Self-refresh, Refresh
Usage Suggestion	PPT retraining is supported as default feature	Optional Suggested if PPT1 retraining time is concern for system performance and if MC can support dfi_ctrlupd_type interface pin	Suggested only if PPT retraining time is concern for system performance AND if MC can support dwc_ddrphy_snoop_en[3:0] interface pin AND LPDDR4X support is not required.

## 7.7 ARC-HS Microcontroller

The microcontroller is an ARC-HS configuration custom generated with a simple set of features to make it very small, yet highly effective for training the pin timing to the off-chip DRAM arrays. The microcontroller core uses a 32-bit Harvard architecture to execute the training application, written in C. The training application controls the pin timing to the DRAMs by writing and reading PHY CSR registers to change pin delays or to launch processes using separate ACSM and DTSM training engines that can iteratively search for the timing edges of the data eye while the C application monitors their results. The training application calculates the final eye centers and applies the optimized values to the PHY CSRs when training completes.

ARC-HS supports two types of closely coupled memories: Instruction Closely Coupled Memory (ICCM) and Data Closely Coupled Memory (DCCM).

ICCM is implemented with two physical memory banks (ICCM0-1), and each bank has identical behavior.

DCCM is implemented with four physical memory banks (DCCM0-3). The DCCM0/1/2/3 on ARC HS have identical behavior.

ARC-HS also uses separate SRAMs for branching and prediction tables.

Example table showing connection between PHY/ARC and external SRAMs are shown in [Table 7-8](#). The “Num of Instances” and “SRAM Instance size” shown in the table is based on test chip implementation. However, customer may implement any other valid combination of SRAMs based on cell availability in their library.

**Table 7-8 Example Connection Between PHY and External SRAMs**

SRAM Name	SRAM Instance Size	Num of Instances	Clock Name (RTL)	Note
ICCM0	3K x 78	2	iccm0_bank0_clk	Connect iccm0_data_din0/iccm0_data_dout0/iccm0_data_addr0/iccm_data_me0/iccm_data_we0/iccm0_data_wem0/iccm0_bank0_clk signals between PHY and ICCM Bank0 SRAM.
ICCM1	3K x 78	2	iccm0_bank1_clk	Connect iccm0_data_din1/iccm0_data_dout1/iccm0_data_addr1/iccm_data_me1/iccm_data_we1/iccm0_data_wem1/iccm0_bank1_clk signals between PHY and ICCM Bank1 SRAM.
DCCM0	3K x 39	2	clk_dccm_bank0_lo	SRAM data is connected to “dccm_bank0_din/dout[0:38]” PHY pins. SRAM WEM is connected to “dccm_bank0_wem[0:4]”PHY pins. Connect dccm_bank0_addr_lo/dccm_bank0_cs_lo/dccm_bank0_we_lo/dccm_bank0_wem_lo/clk_dccm_bank0_lo signals between PHY and DCCM SRAM.

SRAM Name	SRAM Instance Size	Num of Instances	Clock Name (RTL)	Note
DCCM1	3K x 39	2	clk_dccm_bank0_hi	<p>SRAM data is connected to “dccm_bank0_din/dout[39:77]” PHY pins.</p> <p>SRAM WEM is connected to “dccm_bank0_wem[5:9]” PHY Pins.</p> <p>Connect dccm_bank0_addr_hi/dccm_bank0_cs_hi/dccm_bank0_we_hi/dccm_bank0_wem_hi/clk_dccm_bank0_hi signals between PHY and DCCM SRAM.</p>
DCCM2	3K x 39	2	clk_dccm_bank1_lo	<p>SRAM data is connected to “dccm_bank1_din/dout[0:38]” PHY pins.</p> <p>SRAM WEM is connected to “dccm_bank1_wem[0:4]” PHY Pins.</p> <p>Connect dccm_bank1_addr_lo/dccm_bank1_cs_lo/dccm_bank1_we_lo/dccm_bank1_wem_lo/clk_dccm_bank1_lo signals between PHY and DCCM SRAM.</p>
DCCM3	3K x 39	2	clk_dccm_bank1_hi	<p>SRAM data is connected to “dccm_bank1_din/dout[39:77]”</p> <p>SRAM WEM is connected to “dccm_bank1_wem[5:9]”</p> <p>Connect dccm_bank1_addr_hi/dccm_bank1_cs_hi/dccm_bank1_we_hi/dccm_bank1_wem_hi/clk_dccm_bank1_hi signals between PHY and DCCM SRAM.</p>
BC0	256 x 64	1	bc_ram0_clk	Connect bc_din0/bc_dout0/bc_addr0/bc_me0/bc_we0/bc_wem0/bc_ram0_clk signals between PHY and Branch Cache (BC) SRAM.
BC1	256 x 64	1	bc_ram1_clk	Connect bc_din1/bc_dout1/bc_addr1/bc_me1/bc_we1/bc_wem1/bc_ram1_clk signals between PHY and Branch Cache (BC) SRAM.
PT0	1024 x 8	1	gs_ram0_clk	Connect gs_din0/gs_dout0/gs_addr0/gs_me0/gs_we0/gs_wem0/gs_ram0_clk signals between PHY and Prediction Table (PT) SRAM.

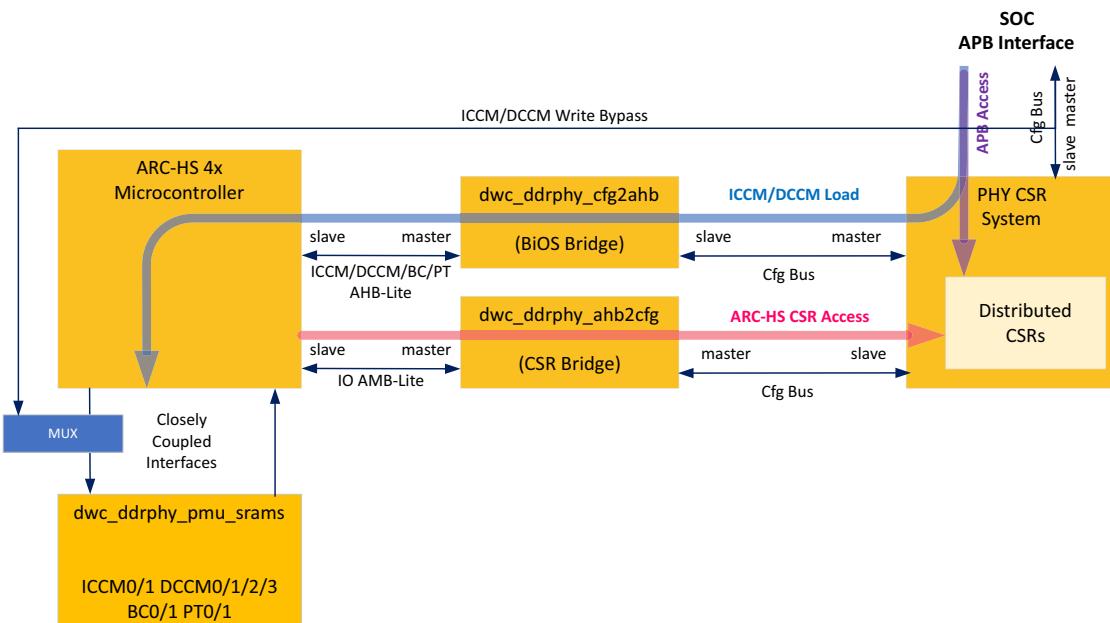
SRAM Name	SRAM Instance Size	Num of Instances	Clock Name (RTL)	Note
PT1	1024 x 8	1	gs_ram1_clk	Connect gs_din1/gs_dout1/gs_addr1/gs_me1/gs_we1/gs_wem1/gs_ram1_clk signals between PHY and Prediction Table (PT) SRAM.

The ARC will run at DfiClk.

The ARC microcontroller is embedded into the DDR PHY design by using bridges to convert the AHB-Lite IO paths to the protocol used internally on the Cfg Bus. A bridge known as the “CSR Bridge” allows code running on the microcontroller to see the PHY CSRs as directly mapped into its memory address space. PHY CSRs, being an array of registers, looks similar to an array of memory locations, albeit distributed throughout the PHY design. In reverse, an APB interface can also control the Cfg bus and can see the ICCM and DCCM microcontroller memories as mapped into PHY CSR address space through the use of the “BIOS Bridge.” Mapping the microcontroller memories into PHY CSR space provides a means of loading the application program into the instruction and data memories that the microcontroller will execute from. Ultimately, both the ARC microcontroller and the APB interface can see all of the PHY CSR registers and all of the ICCM and DCCM microcontroller memory space as a contiguous address space.

Figure 7-9 shows the block diagram of this embedded ARC-HS microcontroller. The red arrow shows the command path for ARC CSR accesses while the blue arrow shows the command path for an SoC host on APB loading a program into ICCM and DCCM memory. The purple arrow shows the APB can issue commands for CSR access as well.

**Figure 7-9 Embedded ARC-HS**



## 7.7.1 ARC-HS Features

The microcontroller is generated with the following basic features:

- ICCM instruction closely-coupled memory populated to 48k bytes each
- DCCM0/1/2/3 data closely-coupled memory populated to 24k bytes each
- An IO port with AHB-Lite interface mapping PHY CSRs into microcontroller memory space
- A pair of AHB-Lite interfaces for ICCM and DCCM initialization of the training C application
- Little Endian memory orientation

## 7.7.2 Mapping of ICCM and DCCM in CSR Address Spaces

**Table 7-9** Mapping of ICCM and DCCM in CSR Address Spaces

Memory Type	PHY CSR Address (16-bit word)
ICCM*	0x005_0000 to 0x005_7FFF
ICCM0	<ul style="list-style-type: none"> <li>■ 0x005_0000 + (4*i)</li> <li>■ 0x005_0000 + (4*i + 1)</li> </ul>
ICCM1	<ul style="list-style-type: none"> <li>■ 0x005_0000 + (4*i + 2)</li> <li>■ 0x005_0000 + (4*i + 3)</li> </ul>
DCCM*	0x005_8000 to 0x005_FFFF
DCCM0	0x005_8000 + (4*i)
DCCM1	0x005_8000 + (4*i + 1)
DCCM2	0x005_8000 + (4*i + 2)
DCCM3	0x005_8000 + (4*i + 3)



- Note**
- The index "I" ranges from 0 to 0x17FF to account for 6K ICCM/DCCM locations to target 96KB of SRAM, ["SRAM"](#) on page 200.
  - PHY can support maximum of 128KB of ICCM/DCCM SRAM (without any design change) if required in future by Training Firmware.

### 7.7.3 ARC SRAM Write Mask Mapping

**Table 7-10 ICCM Write Mask Mapping**

ICCM Write	Mask Mapping
wem[0]	din[7:0]
wem[1]	din[15:8]
wem[2]	din[23:16]
wem[3]	din[31:24]
wem[4]	din[39:32]
wem[5]	din[47:40]
wem[6]	din[55:48]
wem[7]	din[63:56]
wem[8]	din[70:64], ECC for lower 32bit data (unused if no ECC)
wem[9]	din[77:71]. ECC for higher 32bit data (unused if no ECC)

**Table 7-11 DCCM Write Mask Mapping**

ICCM Write	Mask Mapping
wem[0]	din[7:0]
wem[1]	din[15:8]
wem[2]	din[23:16]
wem[3]	din[31:24]
wem[4]	din[38:32], ECC for lower 32bit data (unused if no ECC)
wem[5]	din[46:39]
wem[6]	din[54:47]
wem[7]	din[62:55]
wem[8]	din[70:63]
wem[9]	din[77:71], ECC for higher 32bit data (unused if no ECC)

### 7.7.4 ICCM and DCCM Program Loading

The blue arrow in [Figure 7-9](#) shows the path an SOC host normally uses to load a program into ARC ICCM and DCCM closely-coupled memories. Writes and reads of ICCM and DCCM on this path are supported only when ARC HS4X is stalled. Stall can be applied by writing the StallToMicro bit in the MicroReset register to one at any time, and released into the run state by writing it back to zero. If a new program is

loaded, ARC HS4X can also be reset through the ResetToMicro bit in the MicroReset register to force program execution to start from the beginning.

To speed initialization, the PHY provides a bypass path (top line in [Figure 7-9](#)) for faster loading of the ICCM and DCCM (via faster APB write transactions), as well as a fast zero-fill mechanism for the DCCM SRAM. These are enabled via csrCCMWriteBypassEnable and csrStartDCCMClear respectively. When using the bypass path, all APB Writes to the ICCM and DCCM must be performed in consecutive even/odd pairs, first to an even address followed by the consecutive odd address.

## 7.7.5 ECC Support

The PHY has ECC support enabled. This includes the following:

- Correction of single-bit data errors (ICCM0-1/DCCM0-3)
- Detection of double-bit data errors (ICCM0-1/DCCM0-3)
- Additional detection of double-bit error during exception (trap) handler

The PHY provides performance counters to count corrected errors. Detect-only errors can be enabled to cause interrupts within the PHY Interrupt Controller.

### 7.7.5.1 Notes about ECC support

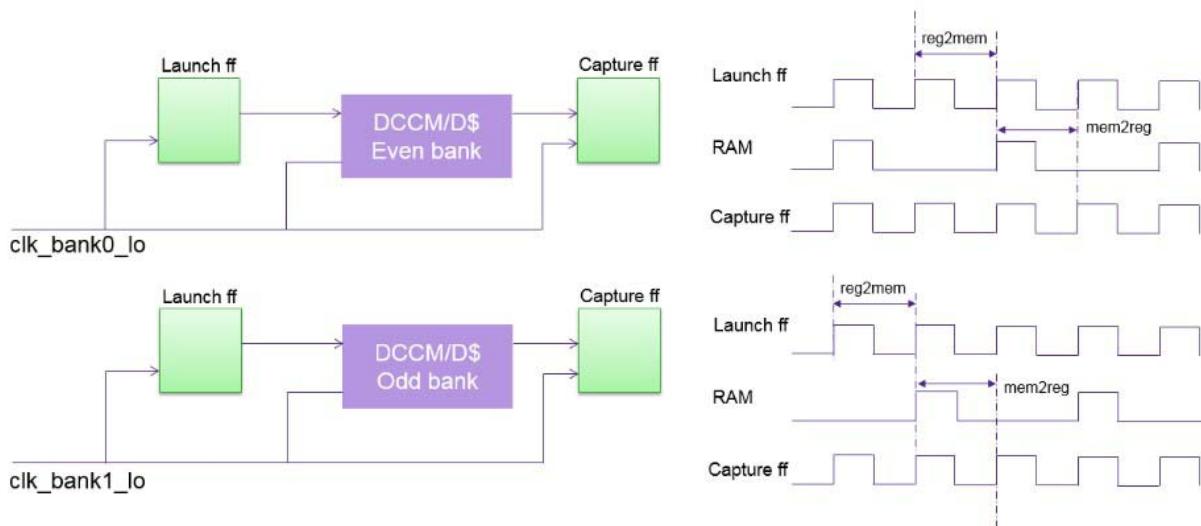
1. ECC can be disabled at run time using csrArcPmuEccCtl.
2. If the memories provided are not wide enough to support ECC, it MUST be disabled.
3. The ARC-HS can take a few cycles to correct a single-bit data error. Thus, if a location is read twice in consecutive (or almost consecutive) cycles, a single-bit data error can be detected more than once.
4. If ECC is disabled, then ARC-HS ECC error (csrPhyEccEn) must be disabled, and the ARC-HS ECC mechanism must be disabled with csrArcPmuEccCtl.

## 7.7.6 ARC and SRAM Clocking

The ARC core runs at UcClk (same as DfiClk) frequency.

PHY provides independent clock outputs for external SRAMs (ICCM/DCCM/PT/BC) which may run at full UcClk frequency or gated UcClk frequency. The ARC and SRAM interface timing must be closed at single UcClk cycle as shown in below timing diagram:

1. Launch and Capture flops, shown with green color in the picture, are inside the ARC.
2. The timing path could be ARC(launch) -> SRAM(capture) or SRAM(launch) -> ARC(capture).
3. Clock for launch and capture is always UcClk.

**Figure 7-10 ARC and SRAM Clocking Diagram**

### 7.7.7 External AHB-Lite Interface



**Note** This interface is reserved for future use and should be ignored. The `csrContextToMicro` must be set to 0.

## 7.8 PHY PCLK DCA

The PHY uses both rising and falling edge of PCLK delayed by LCDL at the transmitter, thus any duty cycle distortion will impact timing margin of transmitted eyes.

### 7.8.1 Duty Cycle Adjuster in PCLK Receiver

PHY performs PCLK Duty cycle correction at the root of PCLK within each DQ lane or AC lane pair in the PCLK Receiver. The table below shows the valid encodings for CSR control of PCLK duty cycle. The codes are represented in 2's compliment format with MSB of coarse representing the sign. Each fine step is approximately 2ps of adjustment, however significant non-linearity relative to PVT is expected, specially with coarse and fine cross-over point.

DCA code updates are glitch-less to PCLK. To propagate a code update, PclkDcaUpdateEn must be pulsed after writing desired PCLK DCA Code.

**Table 7-12 DCA encodings vs Duty Cycle**

	<b>csrPclkDcaCoarse[4:0]</b>	<b>csrPclkDcaFine[3:0]</b>
Decrease Duty Cycle	Reserved	N/A
	Reserved	N/A
	Reserved	Reserved
	Reserved	0xE
	Reserved	0xD
	0x14	0xC
	0x13	0xB
	0x12	0xA
	0x11	0x9
Default	0x00	0x0
Increase Duty Cycle β	0x01	0x1
	0x02	0x2
	0x03	0x3
	0x04	0x4
	Reserved	0x5
	Reserved	0x6
	Reserved	Reserved
	Reserved	N/A
	Reserved	N/A

## 7.8.2 PCLK Duty Cycle Detection

PHY supports duty cycle distortion detection. The table below indicates this based on datarate.

Datarate	Duty Cycle
<= 3200Mbps	DCA disabled.
> 3200Mbps	Use DCD comparator for phase offset detection

When DCD comparator is used, its internal offset needs to be calibrated. This process is done during initial training by Devinit and the results are saved and reused subsequently.

For datarates less than or equal to 3200Mbps, PHY timing budget will reflect any duty-cycle distortion that PHY does not adjust through PClk DCA feature.

## 7.8.3 PCLK DCA Calibration

The PHY supports full search algorithm triggered by PIE during PHY State changes. Full search algorithm is target for the scenario which PHY can be in low power state in long periods and the correction can be larger than 1 fine code. Once full search is used, the search is redone from scratch and all prior search results are ignored.

## 7.8.4 Triggering DCA Calibration for Debug

PCLK DCA calibration can be triggered manually by de-asserting csrPclkDCADynCtrl.PclkDCACalReset. See CSR description of field details. When DCA calibration is complete, PHY will write PclkDCADone. Note: for correct operation, a number of csrPclkDCA\* registers may need different values.

## 7.9 LPDDR5X/5 Specific Features

### 7.9.1 Deep Sleep Mode Support

Deep Sleep Mode (DSM) is an additional Self-Refresh mode with longer Entry/Exit times allowing the DRAM to manage internal circuits for low current consumption (specified by IDD6).

PHY supports Deep Sleep Mode only during LP3/Retention power saving state because it is recommended that Deep Sleep Mode is maintained for a relatively long time to reduce Self Refresh current more efficiently.

Controller must meet the following requirements:

1. Before exiting LP3/Retention :
  - MC must keep memory in DSM state for at least tPDN\_DSM duration.
2. After exiting LP3/Retention:
  - MC must wait for tXSR\_DSM duration after dfi\_init\_complete transitions from 0 -> 1 before sending SRX command.

### 7.9.2 Write X Function Support

Write X function is an optional feature in LPDDR5 device used for reducing power in write operation of repeated data pattern of all 0s or 1s.

Controller must not assert dfi\_wrdata\_en with write X command and dfi\_wrdata input is don't care by the PHY.

Controller can send CAS-WRX command when DRAM is in WCK2CK sync state as well as when DRAM is not in sync state:

- When DRAM is not in WCK2CK synced state :: Controller should assert dfi\_wck\_en with CAS-WRX, similar to CAS-WS\_WR with following restrictions:
  - a. Dfi\_wck\_toggle is STATIC while dfi\_wck\_en is asserted to reduce power. However toggling WCK is also allowed.
  - b. DFI timing parameter Twck\_en\_wrX should be used.
- When DRAM is in WCK2CK synced state :: Controller should extend the WCK state.

To satisfy tWR2WCK(Max) parameter, CSR WCKEXTENSION = 1 is required when DRAM MEMCLK frequency is greater than 688MHz and less than 800MHz with CK:WCK = 1:2 mode.

### 7.9.3 Link ECC Support

LPDDR5 memory supports write and read link ECC. These features are optional and can be enabled by programming MR22.

To enable "Write Link ECC" functionality in PHY, CSR csrWriteLinkEcc should be programmed 1'b1. PUB will transmit dfi\_wrdata\_link\_ecc input on RDQS\_t PHY output during write transaction. The timing for "write link ECC" is same as write data.

RDQS\_c PHY output will be Hi-z during write operation.

### 7.9.4 Strobe-less Read Mode

LPDDR5 memory provide option of not providing read DQS strobe along with read data. This option can be enabled by programming DRAM MR20[1:0]=2'b00 and enabling single-ended mode for CK and WCK.

When strobe-less mode is enabled, retraining is not supported.

In this strobe-less mode:

1. PUB generates RxDigStrobe[7:0] pulses to SE slices to capture the receive data.
  - a. One tWCK toggling preamble ( 2 UI)
  - b. One tWCKtoggling postamble (2 UI)
  - c. BLn toggling strobes ( 16 or 32 UI)
2. RxReplica drift tracking and compensation is disabled.
  - a. csrEnRxClkCor is set to 0.
  - b. csrRxReplicaTrackEn is set to 0.
3. Training steps are as below:
  - a. A new training step (RxDigStrobe Training) will be implemented to training the internally generated strobe and center it within per bit data eye.
  - b. RxEn is not needed to gate external read strobe and will be forced 0 from PUB.
  - c. RxEn training is not required.
4. Drift in tWCKDQO is detected using DRAM oscillator
  - a. Drift Tracking by MRR snooping.
    - This will update the RxDigStrbDly for each lane instead of the RxEnDly.
    - All other training steps remain unchanged.
5. Following are related CSRs:
  - a. csrRxDigStrbEn: per Dbyte; when asserted, Strobe-less read mode is enabled.
  - b. csrTrainEnRxDigStrb: TrainingParam bit 1; when asserted, substitute TrainingCntr for Register RxDigStrbDly during training and during mission mode. Coarse Fields are only available for RxDigStrbDly.
  - c. csrRxDigStrbDly[11:7]: 5 bits per Tg and per PState, used to provide coarse delay in terms of 2UI increment on csrRxDigStrbDly[11:0] output from PUB. Should be programmed N \* 2UI + (tWCKDQO/UI).
  - d. csrRxDigStrbDly[6:0]: 7 bits per Tg and per PState, used to provide fine LCDL delay for RxClk in each DQ lane, in units of 1/64 UI.
  - e. csrRxClkT2UIDly[9:7]: 2 bits for RxClk Coarse delay used by PUB to generate DQ lanes' RXFIFO read pointers. Should be 2'b10 for SkipTrain mode, such as, second posedge DQS\_t strobe captures the first datum.

### 7.9.5 Support LPDDR5 Low Frequency

1. In LPDDR5, In 1:4 Freq Ratio for CK Frequency less than or equal 133MHz (CK freq <= 133)
  - a. Program CsrAcPipeEn = 1
  - b. tctrl\_delay is increased by 1 DfiClk
  - c. twck\_dis, twck\_en\_fs, twck\_en\_rd, twck\_en\_wr is increased by 4 WCK
  - d. tphy\_wrlat, tphy\_wrcslat is increased by 4 WCK
  - e. tphy\_rddata\_en, tphy\_rdcslat is increased by 4 WCK
2. In LPDDR5, In 1:2 Freq Ratio
  - a. For CK Frequency less than or equal 688MHz and greater than 267MHz (267 < CK freq <= 688)
    - i. Program CsrAcPipeEn = 1
    - ii. tctrl\_delay is increased by 1 DfiClk
    - iii. twck\_dis, twck\_en\_fs, twck\_en\_rd, twck\_en\_wr is increased by 2 WCK
    - iv. tphy\_wrlat, tphy\_wrcslat is increased by 2 WCK
    - v. tphy\_rddata\_en, tphy\_rdcslat is increased by 2 WCK
  - b. For CK Frequency less than or equal 267 MHz (CK freq <= 267)
    - i. Program CsrAcPipeEn = 2
    - ii. tctrl\_delay is increased by 2 DfiPubClk
    - iii. twck\_dis, twck\_en\_fs, twck\_en\_rd, twck\_en\_wr is increased by 4 WCK
    - iv. tphy\_wrlat, tphy\_wrcslat is increased by 4 WCK
    - v. tphy\_rddata\_en, tphy\_rdcslat is increased by 4 WCK

### 7.9.6 Single Ended Mode Support

DIFF IOs can be configured to operate in single ended mode, in one or both directions, TX and RX.

- For TX, only one of the IO bumps, \*\_T or \*\_C, will drive data out, CK\_C will be forced 1, while the other bump is forced 0.
- For RX, only one of the IO bumps will be sampled, while the other is ignored. The bump being sampled will have its voltage compared against an internal VREF voltage, provided by an internal DAC in DIFF\_IO ckt.

LPDDR5 single-ended mode:

- When DRAM supports it, PHY can support single-ended mode on CK, WCK, and read DQS.
- Each of these can be independently enabled/disabled by PState-able CSR.
- When Single ended Clock mode is enabled:
  - PHY drives Clock out in BP\_\*\_CK\_T output pins.
  - PHY drives BP\_\*\_CK\_C output pins to 1 using bypass driver.
  - No need for single ended mode RX.
- When Single ended WCK mode is enabled:
  - PHY drives WCK out in BP\_\*\_WCK\_T or BP\_\*\_WCK\_C output pins

- PHY drives BP\_\*\_WCK\_C or BP\_\*\_WCK\_T output pins to 0 using bypass driver.
- When Single ended RDQS mode is enabled:
  - The receive threshold voltage is set in the internal VREF DAC
  - PHY DQS is sampled from BP\_\*\_RDQS\_T or BP\_\*\_RDQS\_C pins
- LPDDR5 DRAMs support single-ended mode only at 1600 Mbps or lower
  - DRAM ODT and NT ODT states for CK, CA, WCK, RDQS, DQ and DMI are required to be unterminated during Single-ended mode for Clock, Write Clock and/or RDQS per JEDEC spec

### 7.9.7 DRAM ZQ Calibration

LPDDR5/5X DRAM supports either Background calibration mode (MR28 OP[5] = 0) or Command-based calibration mode (MR28 OP[5] = 1). ZQ calibration mode can be selected by PHYINIT Userinput ZqMode which will control DRAM command sequence during 1st dfi\_init\_start and LP3 exit.

DRAM ZQ calibration behavior will be as below:

1. ZQ calibration during initial training

PHY is responsible for DRAM ZQ calibration during initial training.

Training firmware always uses MR28 OP[5] ZQ Mode = 0 (default, Background ZQ calibration mode). In each DRAM channel, ZQ calibration is automatically executed using external ZQ register. ZQ calibration sequence is managed by DRAM. Training firmware issues MPC ZQCal Latch command for each channel during Devinit sequence. At the end of initial training, Training FW programs MR28 OP[5] = ZqMode.

2. ZQ calibration during mission mode

Controller is responsible for DRAM ZQ calibration during mission mode.

In case of Background calibration mode (MR28 OP[5] = 0), Controller needs to issue MPC ZQCal Latch command for each channel.

In case of Command-based calibration mode (MR28 OP[5] = 1), Controller needs to issue MPC ZQCal Start command to ZQ Initiator die and issue MPC ZQCal Latch command for each channel. ZQ initiator die is defined in JEDEC package spec. In 4-channel package, ZQ initiator will be rank0 of Channel A die and rank0 of Channel D die.

3. ZQ calibration during 1st dfi\_init\_start/LP3 exit

PHY is responsible for DRAM ZQ calibration during 1st dfi\_init\_start/LP3 exit.

In case of Background calibration mode (MR28 OP[5] = 0), PHY will program MR28 with default value so that DRAM ZQ calibration is restarted by MR28 OP[1] ZQ Stop = 0. PHY will also issue MPC ZQCal Latch command for each channel. It is allowed for controller to change MR28 values before starting mission mode operation. However, MR28 ZQ Mode must be same value as PHYINIT Userinput ZqMode before entering LP3.

In case of Command-based calibration mode (MR28 OP[5] = 1), PHY will issue MPC ZQCal Start command to ZQ Initiator die and issue MPC ZQCal Latch command for each channel.

For MR28 control in LP3 entry/exit, see “[LP3 / IO Retention Enter](#)” on page [162](#), or see “[LP3 / IO Retention Exit](#)” on page [164](#).

### 7.9.8 Per-pin DFE Support

PHY supports per-pin DFE feature of LPDDR5X DRAM when MR0 OP[7] = 1 (Read only).

Limitations exist depending on Phyinit and firmware version. Refer to PhyInit, Firmware application and release notes for those limitations.

Per-pin DFE (MR41 OP[0] = 1) is supported only for one PState (PHYINIT Userinput FirstPState) with highest data rate. MR41 OP[0] = 0 is required for other PStates.

## 7.10 LPDDR4X Specific Features

### 7.10.1 Single Ended Mode Support

DIFF IOs can be configured to operate in single ended mode, in one or both directions, TX and RX.

- For TX, only one of the IO bumps, \*\_T or \*\_C, will drive data out, CK\_C will be forced 1, while the other bump is forced 0.
- For RX, only one of the IO bumps will be sampled, while the other is ignored. The bump being sampled will have its voltage compared against an internal VREF voltage, provided by an internal DAC in DIFF\_IO ckt.

LPDDR4X single-ended mode:

- When DRAM supports it, PHY can support single-ended mode on CK, write DQS, and read DQS.
- Write and Read DQS single-ended modes should be enabled/disabled together by pstat-able CSR.
- When Single ended Clock mode is enabled:
  - PHY drives Clock out in BP\_\*\_CK\_T output pins.
  - PHY drives BP\_\*\_CK\_C output pins to 1.
- When Single ended write DQS mode is enabled:
  - PHY drives BP\_\*\_DQS\_T pins.
  - PHY drives BP\_\*\_DQS\_C pins to 0.
- When Single ended read DQS mode is enabled:
  - The receive threshold voltage is set in the internal VREF DAC
  - PHY DQS is sampled from BP\_\*\_DQS\_T pins
- LPDDR4X DRAMs support single-ended mode only at 1600 Mbps or lower.

### 7.10.2 WDQS Extension

When in LPDDR4X mode, the PHY will drive DQS\_T and DQS\_C differentially to 0 and 1, respectively, before and after a write burst, except during a memory read transaction.

As a result of WDQS extension, write to read/read to write command gap are as below:

- READ-to-WRITE
  - ODT OFF case
    - Same or different rank:  

$$RL + RU(tDQSCK.max/tCK) + BL/2 + RD(tRPST) - WDQS\_on + 2$$
  - ODT ON case
    - Same or different rank:  

$$RL + RU(tDQSCK.max/tCK) + BL/2 + RD(tRPST) - ODTOn - RD(tODTon.min/tCK) + 6$$
- WRITE-to-READ

- ❑ ODT OFF case
  - Same rank:  
 $WL + 1 + BL/2 + RU(tWTR/tCK)$
  - Different rank:  
 $\text{Max}(4, WDQS\_off - (RL + RD(tDQSCK.min/tCK) - RU(tRPRE/tCK)) + 8)$
- ❑ ODT ON case
  - Same rank:  
 $WL + 1 + BL/2 + RU(tWTR/tCK)$
  - Different rank:  
 $\text{Max}(4, WDQS\_off - (RL + RD(tDQSCK.min/tCK) - RU(tRPRE/tCK)) + 8)$

Note: For WDQS\_off value (defined in JESD209-4D, Table 102), 8 tCK should be added in case BL = 32 in the same way as ODTLooff definition (NOTE2 in Table 167)

## 7.11 Impedance Calibration

The PHY incorporates an impedance compensation engine to match a target value of resistances despite variations in Process, Voltage and Temperature. The analog circuits used for this purpose are grouped inside the ZCAL hard macro while the synthesizable portions of the controller sit inside the PUB.

The goal of the compensation engine is as follows:

- To find the binary pull-down calibration code, that adjusts the driver's to accurately match the target DC pull-down resistance.
- To find the binary pull-up calibration code, that adjusts the driver's to accurately match the target DC pull-up resistance.
- Transmit the calibration codes to the IO drivers when an impedance update is requested.

### Overview

The impedance calibrator is composed of the following:

- External precision resistor, RZN;
- custom analog circuits, located inside ZCAL hard macro;
- RTL code, located inside the PUB.

The calibration circuit is configured/monitored through CSR's prefixed with ZCal.

Note: Calibration should be disabled when running at or below 3200 Mbps. This is handled automatically by phyinit/PIE.

PHYINIT has an UserInput DisZCalOnDataRate. By default DisZCalOnDataRate = 1 setting, PHY ZQ calibration will be stopped when frequency change is executed to the destination PState with <=3200Mbps, where PIE will overwrite the pull-up/down code with maximum value. In case DisZCalOnDataRate = 0, PHY ZQ calibration will not be stopped even at <=3200Mbps, and the most recent pull-up/down calibrated code will be used. When frequency change is executed to the destination PState with >3200Mbps, PHY ZQ calibration will be started and PIE will overwrite the pull-up/down code with initial calibrated value which is saved in initial training sequence. [Table 7-13](#) is summary of PHY ZQ calibration control.

**Table 7-13    PHY ZQ Calibration Management by PIE Sequence**

Data rate	DisZCalOnDataRate	PHY ZQ calibration	ZCal code after Frequency change
> 3200Mbps	*	Run	Initial calibrated PU/PD code loaded by PIE
<= 3200Mbps	0	Run	Calibrated PU/PD code (PIE will not change)
	1	Stop	Max. code (0x1ff) forced by PIE

When calibration is enabled, the Calibration Control FSM, located inside dwc\_ddrphy\_zcont.v, will send a signal to the Calibration FSM, inside dwc\_ddrphy\_zcal.v, to start one calibration run.

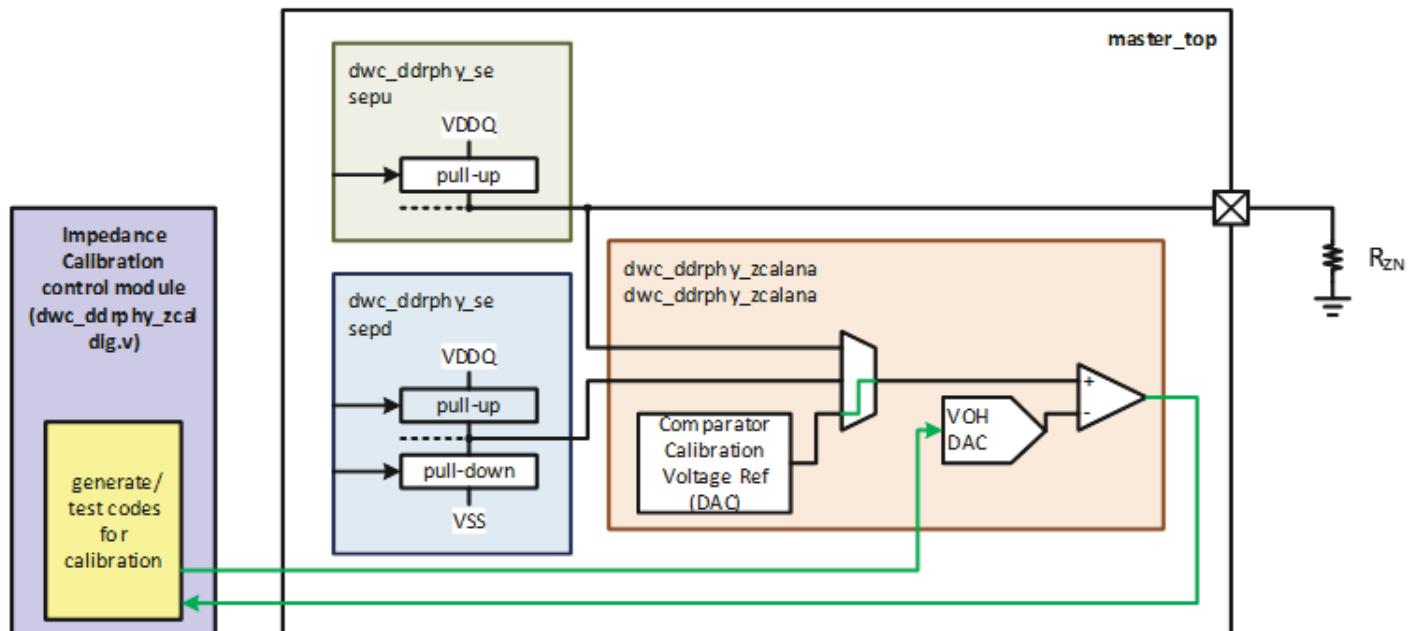
Each calibration run follows the steps below:

1. Enable the analog circuit for calibration.
2. Wait for the analog circuit bias currents to stabilize.
3. Calibrate the comparator, to cancel the comparator's offset voltage.

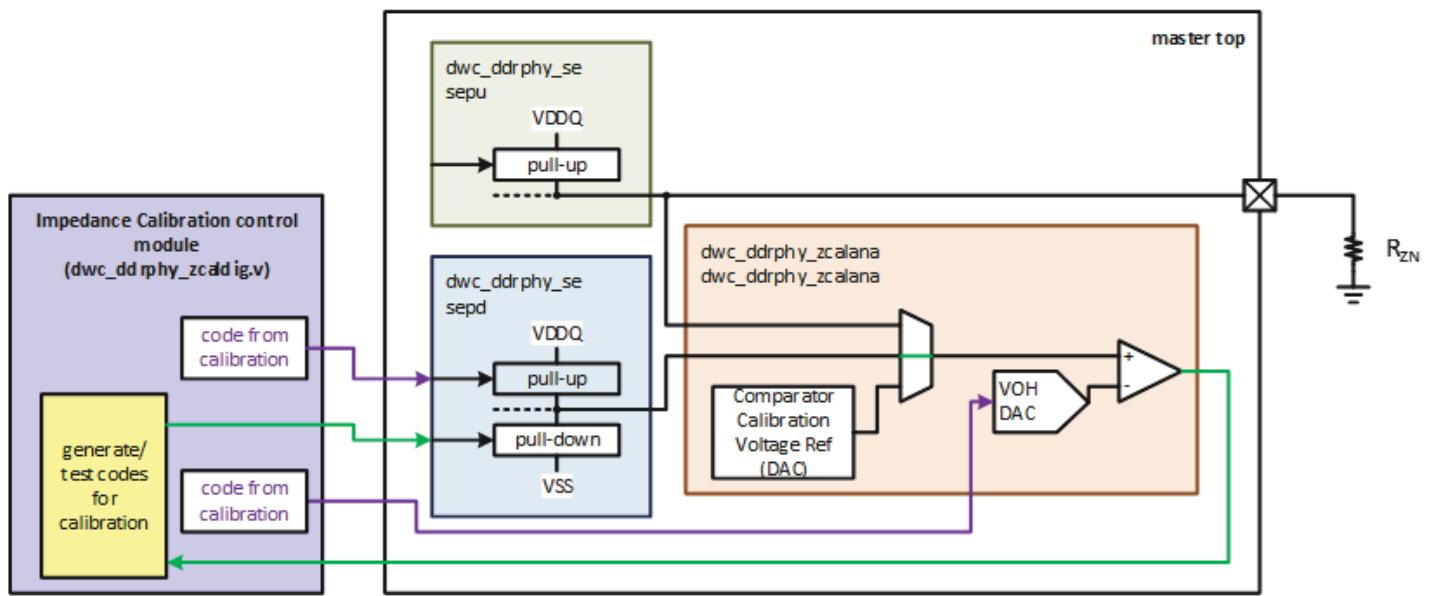
4. Calibrate the pull-up impedance, using one SE slice connected to the external resistor.
5. Calibrate the pull-down impedance, using the second SE slice with the pull-up segments calibrated with value found in the previous step.
6. Disable the analog circuit, used for calibration.

During comparator offset calibration (Figure 7-11), the comparator inputs are connected to a Voltage Reference (VRef) and to a DAC (Digital to Analog Converter). The Voltage Reference is programmed to a target threshold value (VOH). During calibration, jump-search logic will generate different codes for the DAC and will monitor the comparator output. When the DAC output matches the VRef, the offset is canceled, and the comparator calibration is done. The DAC code is saved in a CSR and is used in pull-up and pull-down calibration stages.

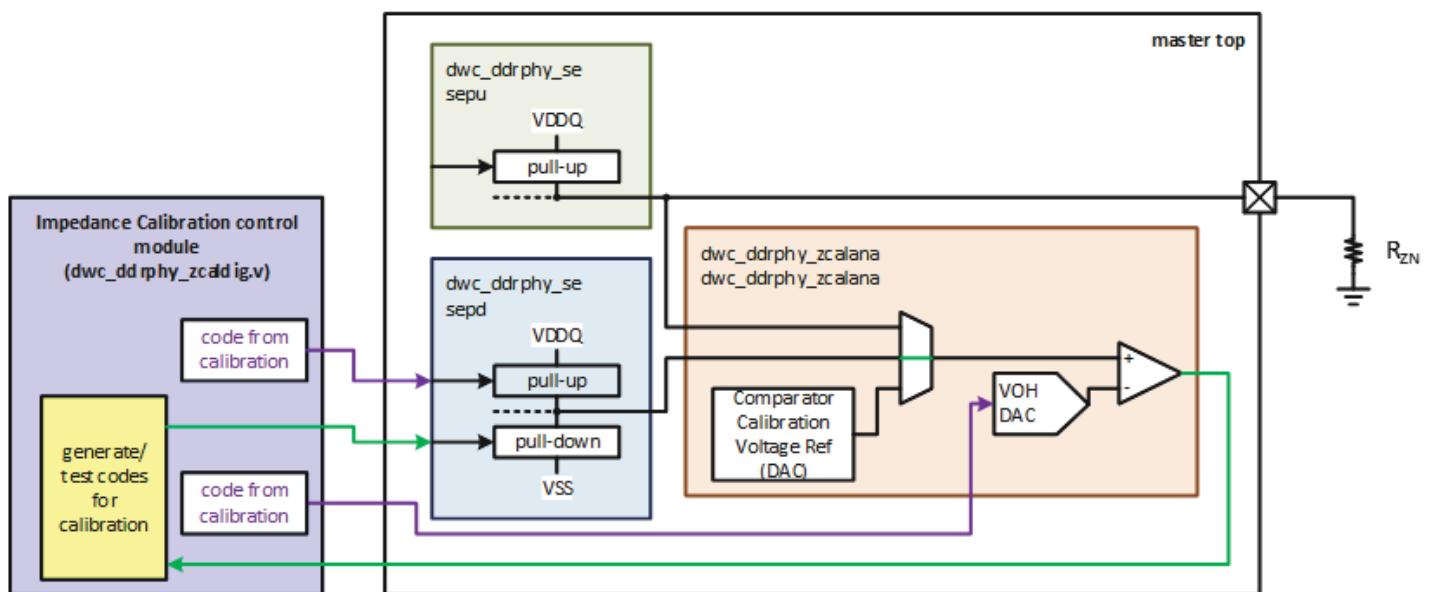
**Figure 7-11 ZCal Comparator Offset Calibration Overview**



During pull-up calibration (Figure 7-11), SEPU replica IO's pull-up is connected to the external precision resistor and connected to the comparator input. The pull-up connected to the external resistor forms a voltage divider. The pull-up target impedance is configured through CSRs. During calibration, jump-search logic will generate different impedance calibration codes for the pull-up and will monitor the comparator output. When the voltage, at comparator input, matches the comparator reference voltage (VOH), the pull-up calibration is done. The pull-up calibration code is saved in a CSR and is used in pull-down calibration stage.

**Figure 7-12 ZCal Pull-Up Calibration Overview**

During pull-down calibration (Figure 7-12), in SEPD IO, both pull-up and pull-down segments are enabled and the node connecting them is connected to the comparator input. The pull-up segment connected to the pull-down segment forms a voltage divider. The pull-up segment is configured for the same target impedance used before, in pull-up calibration, and is configured with the calibration code obtained from the pull-up calibration. During calibration, jump-search logic will generate different calibration codes for the pull-down segment and will monitor the comparator output. When the voltage, at comparator input, matches the comparator reference voltage (VOH), the pull-down calibration is done. The pull-down calibration code is saved in a CSR.

**Figure 7-13 ZCal Pull-Down Calibration Overview**

After one calibration run, the control returns to the impedance control FSM, inside dwc\_ddrphy\_zcalcont.v.

The control FSM may return to the star point and wait for a second trigger event to initialize a new calibration run (csrCalOnce is 1'b1), or the calibration logic may be configured for continuous operation (csrCalOnce is 1'b0), with programmable delay between calibration runs. Non continuous operation may be useful if some sort of VT drift detector exists, that will trigger impedance calibration only when drifts occur. Calibration updates can only be done on DFI sideband events.

After one calibration run is complete, the calibration values are saved in temporary internal registers., and the IOs are updated with the latest values when an update event is received, through a CSR. On every change, a timed signal, cal\_update will be used by the transmitter to capture the new code.

The impedance codes going out to the IO slices could be the calibrated values or override values, which come from override CSRs. An offset value can also be added/subtracted to the calibrated codes going out to the IO slices.

<b>csrOdtSeg120 PU0[3:0]</b>	<b>csrOdtSeg120 PD0[3:0]</b>	<b>csrOdtSeg120 PU1[3:0]</b>	<b>csrOdtSeg120 PD1[3:0]</b>
0001	0000	0001	0001

### 7.11.1 External Reference Resistor

An external precision resistor (RZN), connected between BP\_ZN port and VSS, is required and used as a reference by the impedance calibration circuit. By default, a 120 Ohm, 1% tolerance, resistor is required.

### 7.11.2 Calibration Engine Setup and CSRs

The PHY requires the RZN reference (used for impedance calibration and compensation) to be at the target impedance prior to starting the PHY Impedance Calibration engine. The RZN reference impedance must be stable from this point forward until a reset (or power-cycle) event.

Before a calibration cycle can be started, a few related csr's should be programmed according to the system implementation. These include:

1. ZCalDfiClkTicksPer1uSInfo: The internal counters in the calibration engine uses this register as reference for their various wait times. The engine runs on DfiClk thus this csr should be programmed with the number of DfiClk cycles in 1us. If programmed incorrectly, calibration results may be inaccurate.
2. ZCalOnce: This is used to determine the behavior of impedance calibration runs. If set to 1, the engine will only perform 1 calibration cycle when ZCalRun is asserted. Otherwise it will continuously perform calibration cycles with intervals determined by ZCalInterval.
3. ZCalInterval: This value determines the intervals to perform continuous impedance calibration kickoff.
4. ZCal{Comp,PU,PD}Search{Seed,GainIV,GainTV}Auto: When asserted, enable the usage of previous calibration results as seed values and the usage of a different set of gain values, for the calibrations after the first one. With this mode enabled, is possible to configure the 2nd and next calibrations to take less run time than first calibration do.
5. ZCalDoubleLoopEn: When asserted, enable a second round of pull-up and pull-down calibrations during the 1st calibration run, to improve accuracy.

All changes to the calibration settings must be done only when ZCalBusy=0 and with ZCalRun=0 or ZCalReset=1. Calibration setting changes during mission mode are not supported.



Synopsys recommends keeping all other calibration related CSR's at their default values.

### 7.11.3 Starting Calibration

A calibration cycle is started by writing 1 to ZCalRun CSR or when ZCalReset is deasserted with ZCalRun equal to 1.

The wait time for impedance calibration is dependent on DfiClk Freq, ZCalSettlingTime, ZCal{Comp,PU,PD}SearchGainIV , ZCal{Comp,PU,PD}SearchGainTV and ZCalDfiClkTickPer1uS values and in how far the ZCal{Comp,PU,PD}SearchSeed values are from the target value. It may also be different between the 1st calibration run and the ones that follows. Refer to section Calibration Run Time, for further details.



Before starting a calibration cycle, all related csr's must be setup as required. Once Impedance Calibration is complete, the impedance codes will go out to the IO slices when ZQUpdate CSR bit is pulsed.

Impedance calibration must be performed as part of PHY initialization at data rates that require it, during DevInit. The memory controller is expected to simply wait the expected calibration time until calibration is done before assertion of dfi\_init\_start. No further interaction is required from the memory controller. At low data rates, impedance calibration can be disabled. When this is done, to save power, the clocks to the PUB\_ZCAL and HM ZCAL blocks can be disabled via phyinit.

Impedance calibration must also be requested periodically, as part of Periodic Phase Training.

The Results of the calibration can be queried in ZCal{Comp,PU,PD}Results CSRs.



After the initial training is done, if the residency before the first dfi\_init\_start asserted is long enough so that the system temperature vary more than 20 degree C, it is recommended for the host to trigger an extra full ZQ calibration adopting following sequence in task dwc\_ddrphy\_phyinit\_userCustom\_customPostTrain(). The memory controller is expected to simply wait the expected calibration time until calibration is done before assertion of dfi\_init\_start. No further interaction is required from the memory controller.

1. Host sets csrPState to the correct Pstate given the DFICLK frequency
2. Host triggers the ZCAL FSM by writing csrZCalReset=0, followed by csZCalRun=1.
3. Host waits for completions of ZQ Calibration (20us)
4. Host sets csrZcalReset to 1

The impedance calibrator asserts csrZCalBusy until internal FSM's have run though all the stages of calibration. It is not required to poll csrZCalBusy.

#### 7.11.4 Calibration Run Time

One calibration run is the time between ZCalBusy assertion and ZCalBusy deassertion. It includes:

1. Start time to enable and give time for the calibration analog circuit to stabilize;
2. Comparator offset calibration time;
3. Pull-up calibration time;
4. Pull-down calibration time;
5. End time, to disable the calibration analog circuit and deassert ZcalBusy.

Impedance calibration run time depends on the CSRs below:

- ZCalDfiClkTicksPer1uS
  - Holds the number of DfiClks in 1us (rounded up), with minimum value of 24.
- ZCalDoubleLoopEn
  - When set, enable double-loop calibration in 1st calibration run.
- ZCalSettlingTime
  - Specify the time ZCalana's comparator output takes to settle, after a code change in SEPU, SEPD or VOH DAC.
  - The ZCAL FSM will include this value in the total wait time to read the comparator result every time a code change is made
- ZCal{Comp,PU,PD}SearchSeedAuto
  - When set, enables the use of the last calibration result as the search seed value
  - Faster search is expected when the search seed value is the previous calibration code.
- ZCal{Comp,PU,PD}SearchSeed
  - The seed value impact in the calibration time affects mostly the time taken by the jump search logic to move from the start gain to start gain + 1. After this moment, the maximum search time is deterministic and depends on the difference between start gain and end gain.
- ZCal{Comp,PU,PD}SearchGain{IV,TV}Auto
  - When set, enables the use of a different set of search gains between 1st calibration and the ones that follow.
  - Two different run times can be obtained by using different search settings
- ZCal{Comp,PU,PD}SearchGain{IV,TV}Val{,B}

#### 7.11.5 Configuring IO Impedances

The IO drive and ODT impedances codes are configurable through CSRs. They are specified per slice function and not per individual IO (see [Table 7-14](#)). One individual set of TxStrenCode\*/ OdtStrenCode\* codes exists as listed below:

- One code set per Dx4/Dx5 HM for DQ and DMI IOs in that nibble.
- One code set per Dx5/Dx5 HM for DQS{T,C} and WCK [T,C] IOs in that nibble.
- One code set per ACX2 HM for any CA, DTO IOs in that AC pair.
- One code set per CKX2 for any CK[T,C] IOs in that CK pair.

- Notice CSX2 (LP5-CS, LP4X-CKE ) and CMOS HM are uncalibrated.

**Table 7-14 IO Slices Impedance Configuration**

CSRs where user specifies the impedance value for the IO		Impedance Values	Use in	
CSR Name	CSR Field	LPDDR5	DBYTE	AC
TxImpedanceDq	TxStrenCodeDqPU[3:0]	40,60,120, HiZ	DQ/DMI	
TxImpedanceDq	TxStrenCodeDqPD[7:4]	40,60,120, HiZ	DQ/DMI	
OdtImpedanceDq	OdtStrenCodeDqPU[3:0]	40,60,120, HiZ	DQ/DMI	
OdtImpedanceDq	OdtStrenCodeDqPD[7:4]	40,60,120, HiZ	DQ/DMI	
TxImpedanceAC	TxStrenCodePUAC[3:0]	40,60,120, HiZ		AC/CK
TxImpedanceAC	TxStrenCodePDAC[7:4]	40,60,120, HiZ		AC/CK
OdtImpedanceAC	OdtStrenCodePUAC[3:0]	40,60,120, HiZ		AC/CK
OdtImpedanceAC	OdtStrenCodePDAC[7:4]	40,60,120, HiZ		AC/CK
TxImpedanceDqs	TxStrenCodeDqsPU{T,C}[7:0]	40,60,120, HiZ	DQS/WCK	
TxImpedanceDqs	TxStrenCodeDqsPD{T,C}[15:8]	40,60,120, HiZ	DQS/WCK	
OdtImpedanceDqs	OdtStrenCodeDqsPU{T,C}[7:0]	40,60,120, HiZ	DQS/WCK	
OdtImpedanceDqs	OdtStrenCodeDqsPD{T,C}[15:8]	40,60,120, HiZ	DQS/WCK	

## 7.12 Low Power Design

The PHY may reduce its power through selective static settings of CSRs such as <Drive Strength>, <ODT Strength>, <DFE Enable>, <DBYTE Enable>, <AC lane enable>, etc., based on system configuration requirements.

CSR Name	Default Value	Per PState	Notes
LpDqPhaseDisable	5'h0	No	Refer to Register description section for details
AC Lane Controls			
AcLnDisable	12'b0	No	Refer to Register description section for details
DfiClkAcLnDis	12'b0	No	Refer to Register description section for details
PClkAcLnDis	12'b0	No	Refer to Register description section for details
AcRxPowerDownLnX	2'h2	No	Refer to Register description section for details
Dbyte Lane Controls			
DByteDisable	1'b0	No	Refer to Register description section for details
RxDigStrbEn	1'b0	Yes	Refer to Register description section for details
DxDfiClkDis	11'h0	Yes	Refer to Register description section for details
DxPClkDis	11'h0	Yes	Refer to Register description section for details
DxRxStandbyEn	1'b0	No	Refer to Register description section for details
RxPowerDownLnX	1'h0	No	Refer to Register description section for details
LpDqPowerDnEn	1'h0	No	Refer to Register description section for details

Dynamic frequency scaling may be used to conserve power by operating at only the minimum frequency required by each system operating state. Performance tuning may be tailored independently to each frequency set point (FSP) to ensure robust operation with minimal power consumption at each frequency in the PHY and in the DRAM.

DFI Low Power interface (dfi\_lp\_ctrl\_req/ack and dfi\_lp\_data\_req/ack) and dfi\_dram\_clk\_disable can be used by the controller to further save power dynamically. See “[DFI Low Power Interface](#)” on page [316](#) for details.

## 7.13 Support of Dynamic Voltage Scaling

PHY supports dynamic voltage scaling of its internal power rails in two scenarios:

1. Initial power-up (VDD, VDDQ, VAA and VDDQ\_VDD2H)
2. LP3/IO Retention (VDD, VDDQ, VAA).

### 7.13.1 PHY Core DVFS Support



**Note** For automotive applications scaling of core VDD is not supported by the PHY.

The PHY supports scaling of VDD with frequency under the following conditions:

- Core VDD level can be changed during PState change sequence only
  - Core VDD scaling is not allowed while mission mode traffic is on
- Each PState will have its own VDD level and must be trained at its target VDD
- Refer to the “Recommended Operating Conditions” section of PHY Databook for the supported Core VDD, operating frequency ranges, DFI Ratios and protocol dependencies.

### 7.13.2 DRAM DVFSC Support

PHY supports DVFSC using the DFI Frequency Change procedure using `dfi_init_start/dfi_init_complete` interface if DRAM device supports the DVFSC feature.

Each DVFSC state is assigned to a PHY Power State (PState). See “[DFI Frequency Change](#)” on page [171](#) for additional details. Maximum data-rate supported in DVFSC PState is 1600 Mbps, as specified by JEDEC.

### 7.13.3 DRAM DVFSQ Support

PHY supports DRAM DVFSQ ( $VDDQ = 0.3$  V) using the DFI Frequency Change procedure using `dfi_init_start/dfi_init_complete` interface if DRAM device supports the DVFSQ feature.

Each DVFSQ state is assigned to a PHY Power State (PState). PHY IO ZQ Calibration will be disabled when  $VDDQ = 0.3$  V.

PHY can support VDDQ ramp without stopping mission mode (read/write) traffic with following restrictions:

- Current PState MR19.DVFSQ=1 and data rate  $\leq 3200$  Mbps
- Voltage ramp rate should be less than JEDEC defined VDDQSR1/VDDQSR2 slew rate parameter.

Note: Based on limited test chip silicon validation and DRAM vendor feedbacks, it is our understanding that VDDQ change does not impact DRAM tWCKDQI or tWCKDQO timing parameters. Therefore, periodic retraining is not required. Confirm with your memory vendor.

PState/Frequency change is not allowed while SOC is ramping VDDQ up or down. SOC must perform retrain+relock sequence with  $VDDQ = 0.5$  V to change PState/Frequency.

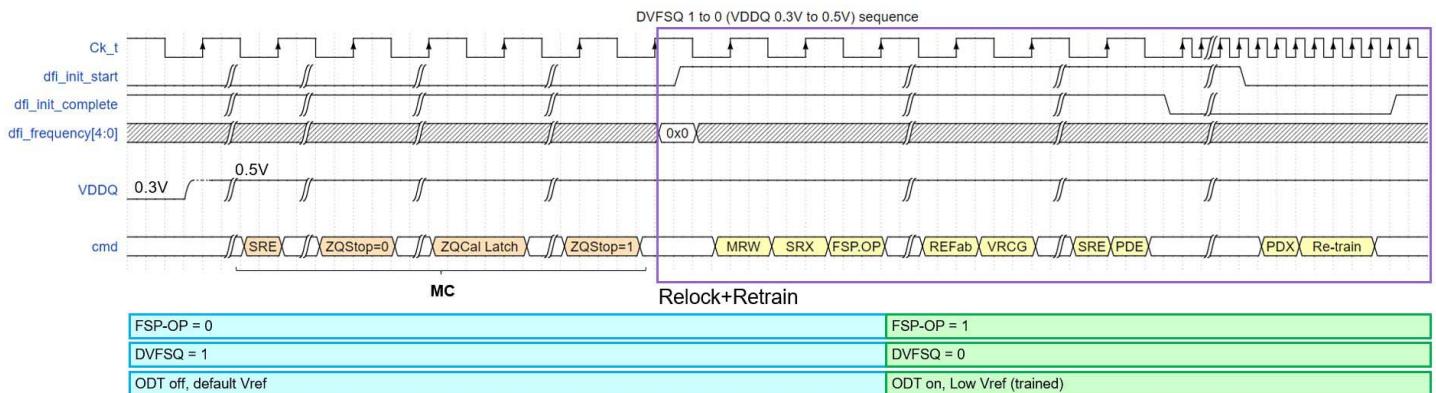
JEDEC requires DRAM background calibration stopped when DVFSQ is active ( $VDDQ = 0.3$  V).

For DVFSQ 1 to 0 transition, VDDQ change is allowed before frequency change.

After VDDQ transition:

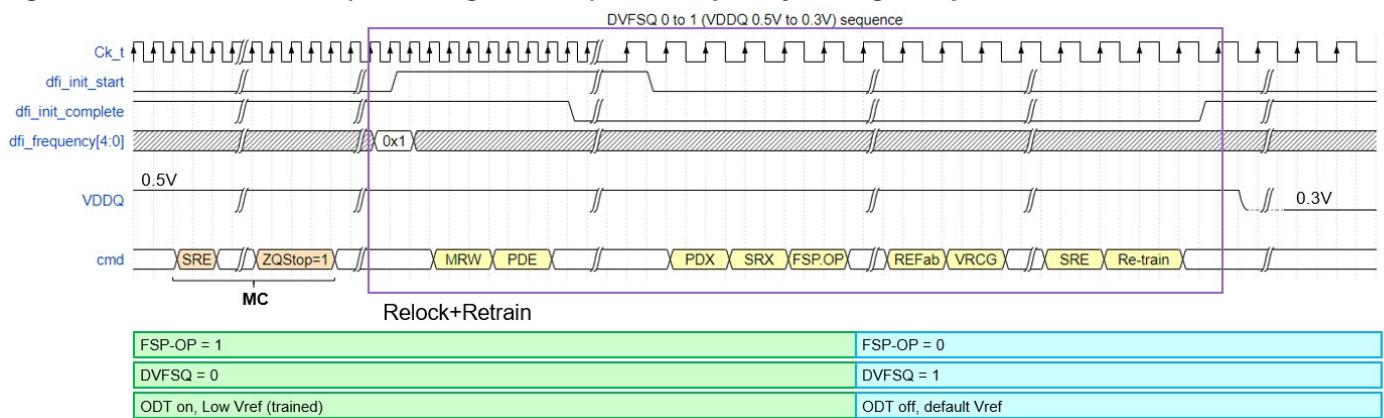
1. MC must issue SRE, MRW (MR28.ZQStop=0), MPC ZQCAL Latch and MRW (MR28.ZQStop=1) commands before asserting dfi\_init\_start. In case Command-based calibration mode (MR28.OP[5] = 1) is selected for DRAM, MC is required to issue ZQCAL Start command before ZQCAL Latch. Instead, MC does not need to change MR28.ZQStop.
2. PHY ZQ calibration must complete (20 us) after VDDQ transition and before asserting dfi\_init\_start. MC can continue mission mode traffic during this period.
3. Frequency change is allowed in dfi\_init\_start/complete sequence with Relock+Retrain. To restart background ZQ calibration in DRAM, MC is required to issue MRW (MR28.ZQStop = 0) after Relock+Retrain before starting mission mode operation.

**Figure 7-14 DVFSQ 1 to 0 (VDDQ Low to High) and Frequency Change Sequence**



For DVFSQ 0 to 1 transition, MC must issue SRE and MRW (MR28.ZQStop=1 after tZQRESET is expired from ZQReset = 1) before asserting dfi\_init\_start. Only frequency change is allowed in dfi\_init\_start/complete sequence with Relock+Retrain. VDDQ change is allowed after frequency change. Execution order of MR16.VRCG = 0 and MR28.ZQ Stop/ZQ Reset is don't care for DRAM since VRCG controls only Vref(DQ/CA) current generators, meaning no impact to ZQ calibration.

**Figure 7-15 DVFSQ 0 to 1 (VDDQ High to Low) and Frequency Change Sequence**



## 7.14 Periodic Phase Training 2 (PPT2)

To use PPT2 re-training feature, user must set PHYINIT userInputAdvanced.RetrainMode.[Pstate]=4 and dfi\_ctrlupd\_req with dfi\_ctrlupd\_type = 2'b01. The difference in ppt2 from ppt1 (Periodic Phase Training) is summarized in [Table 7-7](#) on page [209](#). It is allowed to enable both PPT1 and PPT2.

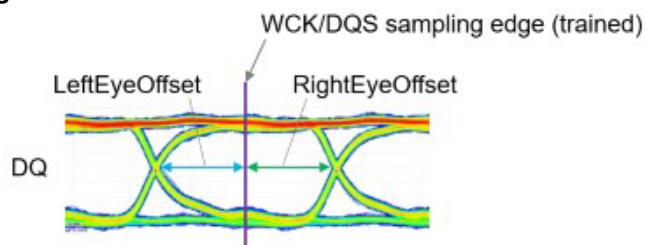
### 7.14.1 PPT2 RxEn Retraining

In this step PHY generates a Read DQS pattern to fine tune its internal read gate signal. As part of initial frequency change training results from csrRxEnDly\*Tg\* would have already been loaded in the internal register RxEnDlyNow. This register is observable via csrDbCurrentDlyTimingInfo by selecting {useRxEnDlyTg\*[10:6],useRxEnPhaseD\*[5:0]} in csrSelDbCurDlyTmngInfo. During each execution of this step, RxEnDlyNow is updated by at most +/-1 fine code according to retrain results. In LPDDR4 this is done in via MPC RD DQ commands; In LPDDR5 with RDC commands.

### 7.14.2 PPT2 TxDq Wr Retraining

In this step PHY fine tunes the value of TxDqDly used for each lane to achieve better eye centering given TxDqs/WCLK timing. As part of initial frequency change training results from csrTxDqDly\*Tg\* would have already been loaded in the internal register TxDlyNow. This register is observable via csrDbCurrentDlyTimingInfo by selecting {useTxDqDlyTg\*[10:6],useTxDqPhaseD\*[5:0]} in csrSelDbCurDlyTmngInfo. During each execution of this step, TxDqDly is updated by at most +/- 1 fine code for each right/left edge check. During initial training, eye left and right edge offsets as defined in [Figure 7-16](#) are saved to csrTxDqLeftEyeOffsetTg\* and csrTxDqRightEyeOffsetTg\* for a simple data pattern used for retraining. During incremental retraining PHY sends WR FIFO and RD FIFO commands to test for the right and left edges of the eye on lane 0 using earlier saved offsets. If both left and right edge samples confirm a drift in the same direction, an adjustment to TxDlyNow is made accordingly.

**Figure 7-16 DQ eye left and right offset**



### 7.14.3 PPT2 LPDDR5 TxDQ Wr ECC Retraining

In LPDDR5 write link ECC is driven on RDQS\_t signals which has the same timing as DQ lanes in the DRAM. As a result an incremental training step is needed to also train this parameter. As part of initial frequency change training results from csrTxDqsDly\*Tg\* would have already been loaded in the internal register TxDqsDlyNow. This register is observable via csrDbCurrentDlyTimingInfo by selecting {useTxDqsDlyTg\*[10:6],useTxDqsPhaseD\*[5:0]} in csrSelDbCurDlyTmngInfo. The algorithm used here is identical to TxDq Wr Retraining in previous section. During initial training, eye left and right edge offsets are saved to csrTxDqsLeftEyeOffsetTg\* and csrTxDqsRightEyeOffsetTg\* for a simple data pattern used for retraining.

In case WrLinkECC is disabled in LPDDR5 mode, TxDQ retraining for RDQS\_t pin is skipped.

#### 7.14.4 PPT2 RxClk Retraining

In this step PHY fine tunes the value of RxClkDly used for each lane to achieve better eye centering. As part of initial frequency change training results from csrRxClkDly\*Tg\* would have already been loaded in the internal register RxClkDlyNow. This register is observable via csrDbCurrentDlyTimingInfo by selecting {useRxClkDlyTg\*[10:6],useRxClkPhaseD\*[5:0]} in csrSelDbCurDlyTmngInfo. The algorithm used here is similar to TxDq Wr Retraining in previous section. During initial training, eye left and right edge offsets are saved to csrRxClkT/CLeftEyeOffsetTg\* and csrRxClkT/CRightEyeOffsetTg\* for a simple data pattern used for retraining.

#### 7.14.5 PPT2 Supported Retraining Limits

- Retraining supports max temp drift from -40 degree C to 125 degree C.
- Retraining supports maximum voltage drift of 50mV.
- Retraining is not supported for data rates < 1600 Mbps.
- It is not allowed for MC to issue dfi\_ctrlupd\_req(type1) when data rate is < 1600Mbps.
- Retraining is not supported in LPDDR5 strobe-less read mode.

## 7.15 Incremental Retraining Used in PPT2

### 7.15.1 Overview

Incremental retraining can be used to compensate for small amount of DRAM Drift during DRAM REFRESH wait periods tRFCab/tRFCpb. The MC must ensure tRFCab/tRFCpb and the PHY latency to complete the training step are satisfied. See dfi\_ctrlupd\_req timing table for exact duration of each type of training. PHY implements an internal sequencer to step through each training step automatically without requiring any communication from MC. To fully compensate for DRAM drift, MC must ensure dfi\_ctrlupd\_req events take place as frequent as needed according system conditions.

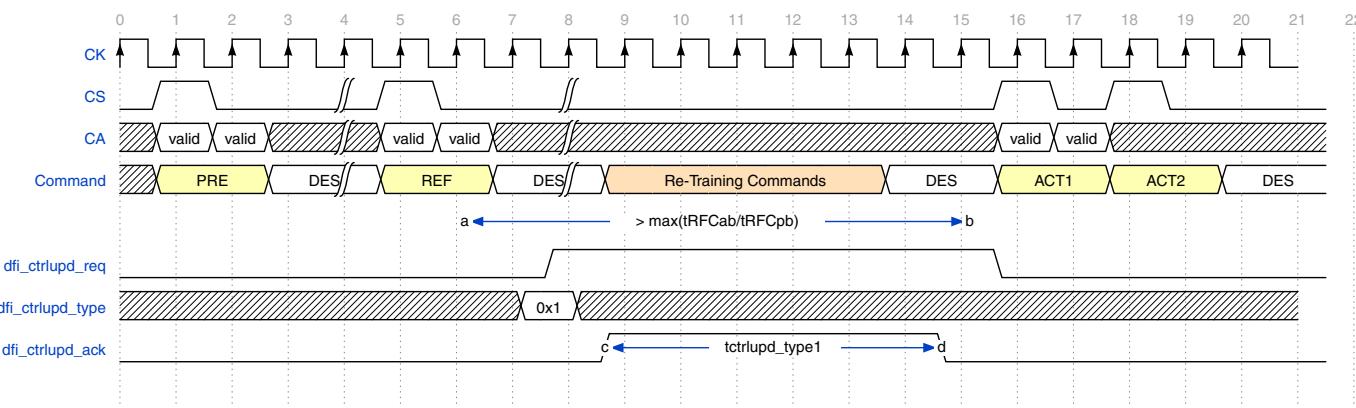


PPT2 algorithms rely on previous execution of full PPT1 retraining at the current PState.

### 7.15.2 Triggering Incremental Retraining

Incremental re-training can be requested via the DFI CTRL update interface. For optimal performance, the memory controller(MC) must issue dfi\_ctrlupd\_req with dfi\_ctrlupd\_type = 0x1 immediately after REFab/REFpb commands. Once dfi\_ctrlupd\_req is issued, MC must wait for PHY to complete the retraining step before issuing next DRAM command (ACTIVE or further REFRESH). The diagram below demonstrates sequence of events for single step of incremental retraining.

**Figure 7-17 LPDDR4 Incremental Retraining During DRAM REFRESH Command**



MC must wait for de-assertion of dfi\_ctrlupd\_ack to ensure DRAM state is preserved. If dfi\_ctrlupd\_req is de-asserted prior to de-assertion of dfi\_ctrlupd\_ack, PHY cannot guarantee correct DRAM state.

Since tRFab/tRFpb are relatively short, only select timing parameters are re-trained on each dfi\_ctrlupd\_req. The PHY contains a programmable sequencer that automatically steps through each set of parameters for each rank. To retrain both ranks the MC must issue enough DFI control update events to cycles through all the steps for a given rank and channel. So to train channel B, MC can issue dfi\_ctrlupd\_req to channel B only after all steps for channel A are complete. The diagram below demonstrates the required sequence of DFI Control update events for correct operation.

When MC asserts dfi\_ctrlupd\_req with dfi\_ctrlupd\_type[1:0]=2'b01, PHY will perform LCDL calibration, followed by one step of retraining. Before asserting dfi\_ctrlupd\_req, MC should de-assert dfi\_wck\_en because WCK needs to be stopped. Here, dfi\_ctrlupd\_req can be asserted at the following DRAM state:

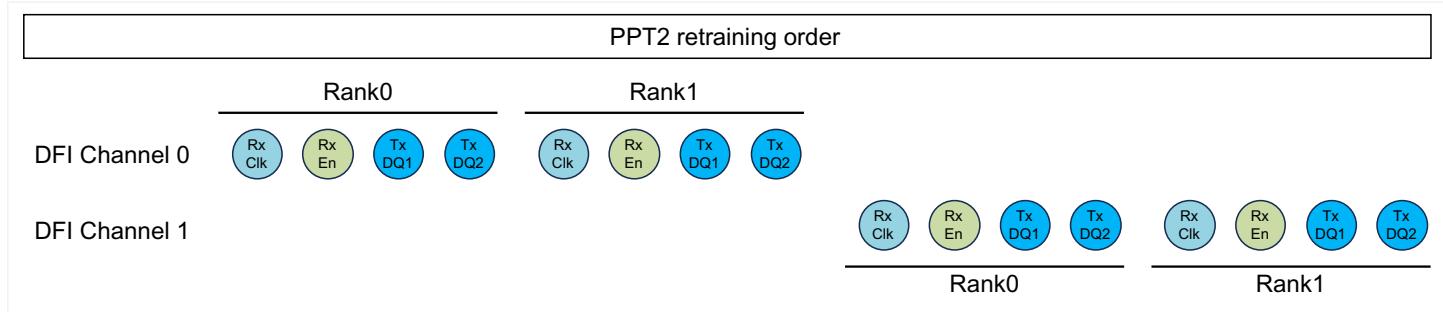
- Idle
- Self-Refresh
- Refresh

The DRAM state is same at entry and exit of ppt2 re-training. PHY stops WCK toggling by CAS(WS\_off) command before exiting each ppt2 re-training. By periodically executing PPT2 using dfi\_ctrlupd\_req with dfi\_ctrlupd\_type = 1, MC is not required to assert dfi\_ctrlupd\_req with dfi\_ctrlupd\_type = 0. This is because ctrlupd\_type = 0 is for LCDL VT-update and this is included in ctrlupd\_type = 1.

Default sequence order is RxClk (Rank0) -> RxEn(Rank0) -> TxDQ1(rank0) -> TxDQ2(Rank0) -> RxClk(rank1) -> RxEn(rank1) -> TxDQ1(rank1) -> TxDQ2(rank1), where TxDQ1 refers to retraining of TxQ/DMI and TxDQ2 refers to retraining of RDQS\_t (WrLinkECC).

PPT2 execution order is reset by dfi frequency change sequence except Retrain only. After dfi frequency change sequence, MC should restart from channel 0, which triggers RxClk ppt2 re-training in rank0.

**Figure 7-18 Suggested Control Update Sequence Order for 2 Rank Configurations with 2 DFI Channels**



- In 2 channel configurations, MC must ensure that dfi\_ctrlupd\_req for both channels are not overlapping and steps 1 to N for Channel A is complete before Channel B ctrlupd type 1 requests are issued.
- MC is required to send dfi\_ctrlupd\_req (type1) to other channel Y after completing RxClk/RxEn/TxDQ1/TxDQ2 PPT2 for both ranks in channel X. MC is not allowed to trigger RxClk PPT2 from rank0 again within same channel X before executing all PPT2 for both ranks in channel Y.

### 7.15.3 Determine Retraining Interval

The minimum retraining interval for each step (TxDQ, RxClk and RxEn) must be determined separately. Retraining step interval is determined by following factors:

- Thermal drift characteristics of the system e.g. 4 degC/sec.
- Thermal drift characteristics of each component e.g. LPDDR4 tDQS2DQ drift rate is 0.6ps/degC.
- LCDL delay step size, for example, 3ps.

Example Calculation of LPDDR4X TxDQ retraining\_interval:

- = (delay tap size / tDQS2DQ drift rate)/thermal drift

- $=[\{3\text{ps} / (.6\text{ps}/\text{degC})\} / 4 \text{ degC/sec}]$
- $= 1.25 \text{ sec}$

In general, considering both temperature and voltage drift, the interval is expressed as:

Interval = (Delay tap size [ps]) / (Thermal drift rate [ps/degC] \* Thermal drift [degC/sec] + Voltage drift rate [ps/mV] \* Voltage drift [mV/sec])



**Note** It is recommended that MC selects optimal retraining\_interval based on minimum calculated interval among all steps.

#### 7.15.4 Determine DFI Control Update Interval

DFI Update interval is defined as minimum time between two dfi\_ctrlupd\_req from the controller.

For 1 rank, 2 channel system in LPDDR5 mode with link ECC enabled:

- MC needs to send total 8 dfi\_ctrlupd\_req to complete RxClk, RxEn, TxDQ1 and TxDQ2 retraining steps.
- DFI Tctrlupd\_interval = 1/8 of retraining\_interval.

For 2 rank, 2 channel system in LPDDR5 mode with link ECC enabled:

- MC needs to send total 16 dfi\_ctrlupd\_req to complete RxClk, RxEn, TxDQ1 and TxDQ2 retraining steps for rank0 and rank1 both.
- DFI Tctrlupd\_interval = 1/16 of retraining\_interval.

Customer may choose any other combination and may calculate required DFI tCtrlupd\_interval timing accordingly.

#### 7.15.5 PHY Retrain Sequencer Options

See PhyInit HTML documentation for the following parameters.

- useInputAdvanced.RetrainSeqOrder

#### 7.15.6 Incremental Retraining Limitations

**Table 7-15 Incremental Retraining PPT2(RetrainMode=4) Limitations**

Item	LP5 or LP4	Working condition	Note
Pstate	common	Pstate P0-P3	
DFI freq. ratio	common	1:4 and 1:2 both modes	
Frequency	common	LP5-6400, 5500, 4266, 3200, 1600 LP4-4266, 3200, 1600 LP5X-9600, 8533, 7500	PPT2 is not supported for < 1600 Mbps.

Item	LP5 or LP4	Working condition	Note
DRAM Write latency	common	Both SetA or SetB	To realize PPT2 runtime < 500ns at highest data rate, WL(setA) is mandatory for data rates > 3200 Mbps, DFI 1:4 or 1:2 To realize PPT2 runtime < 1000ns at highest data rate, WL(setA) is mandatory for data rates > 1600 Mbps and < 3200 Mbps, DFI 1:4 or 1:2
DRAM ECC	LP5	Both WECC and RECC are enabled or Both WECC and RECC are disabled	For RDQSt ppt2, both WECC and RECC need to be enabled

### 7.15.7 Incremental Retraining Timing Parameters

Following table describes tctrlupd\_type1 latency for single PPT2 execution, where the runtime depends on data rate, number of PStates, and DRAM's WL. In case of single PState configuration, only the PState X row is applied. Under multiple PStates configuration, the PState X row is valid only for the PState with highest data rate and the Pstate Y row is applied for the rest of PStates with not highest data rate.

**Table 7-16 tctrlupd\_type\_1 latency for WL(setA)**

Data rate	Pstate X (highest data rate)	Pstate Y (not highest data rate)
3200Mbps <= DR <= 9600Mbps	500ns	750ns
1600Mbps <= DR < 3200Mbps	1000ns	1500ns

Notes:

- When NumPState = 3 (P0 = 8533Mbps, P1 = 3200Mbps, P2 = 1600Mbps, tctrlupd\_type1 = 500ns(P0), 750ns(P1), 1500ns(P2))
- When NumPState = 3 (P0 = 3733Mbps, P1 = 3200Mbps, P2 = 1600Mbps, tctrlupd\_type1 = 500ns(P0), 750ns(P1), 1500ns(P2))
- When NumPState = 1 (P0 = 3200Mbps, tctrlupd\_type1 = 500ns(P0))
- For Pstate using WL(setB), tctrlupd\_type1 value is increased by 50ns

# 8

## System Integration

---

This section provides information for integrating the DDR PHY with either the Synopsys IP Protocol or Memory Controllers or a non-Synopsys memory controller. Depending on the user design and implementation methodology, the top-level PHY Verilog module (DDRPHY\_top) can be instantiated in the chip logic. This top-level PHY Verilog includes the instantiation of all PHY hard and soft IP blocks and therefore greatly simplifies the connectivity of the DDR PHY IP to the user logic.

If the user design methodology and implementation does not allow for the instantiation of the top-level PHY Verilog module, then the two PHY components can be instantiated separately. The PHY soft IP is the PUB, while all PHY hard macros are grouped together under the DDRPHY hierarchy.

This chapter describes the integration when using either method. This chapter describes the following:

- “[DDR PHY Top Level \(DDRPHY\\_TOP\)](#)” on page [244](#)
- “[Additional, Optional Customer Defines](#)” on page [246](#)

## 8.1 DDR PHY Top Level (DDRPHY\_TOP)

### 8.1.1 PHY Configuration

The PHY is highly configurable, making it suitable for a wide range of system architectures. Most of this configuration is implemented using pre-processor `define variables. However, you should use the core Consultant or coreAssembler (GUI or batch mode) tool to configure and generate the PUB RTL source code.

While coreConsultant is the basic tool used to create a “workspace” for a single component, coreAssembler enables you to work with a component within the context of a subsystem. A workspace is your working version of a Synopsys IP component. The CoreConsultant and core Assembler GUI interfaces enable you to run basic sanity testing using the provided System Verilog Verification environment or Customer Test Bench(CTB). You can also perform Synthesis using these interfaces. A full test environment and tests are also supplied.

The recommended approach to configuring the PHY is to use CoreTools. After configuring the PHY using the coreConsultant tool, the parameters are defined in the output file dwc\_lpddr5xphy\_VDEFINES.v. Nevertheless, the below information is provided as a reference guide for those who want to know more about the `define variables used in PHY, or in rare cases where manual configuration is required.

The following `define values are provided as part of the RTL release to customers in the dwc\_lpddr5xphy\_{config}\_VDEFINES.v file. The {config} is determined by the example testbench and customers desired configuration.

The list of all valid and supported configuration is shown in [Table 8-1](#).

Do not set any `defines unless they are explicitly documented in this databook

The contents of the defines are modifiable, with some restrictions, by the customer, with recommended values provided as a unit in several example VDEFINES.v files. The list of defines, and description, follows:

- Number of channels: determined by setting one of the following:
  - DWC\_LPDDR5XPHY\_NUM\_CHANNELS\_1, or
  - DWC\_LPDDR5XPHY\_NUM\_CHANNELS\_2
- BP.DTO0/1 IO pads will be instantiated by setting DWC\_LPDDR5XPHY.DTO\_ENABLED.
  - PHY always supports LPDDR4X, LPDDR5 and LPDDR5X functionality.

**Table 8-1 Supported Configuration**

Supported DFI Channel Define	DWC_LPDDR5XPHY_DTO_EN_ABLED	Define File CONFIG
DWC_LPDDR5XPHY_NUM_CHANNELS_1	undefined	lp5xcs2dq18ch1
DWC_LPDDR5XPHY_NUM_CHANNELS_1	defined	lp5x4xcs2dq18ch1
DWC_LPDDR5XPHY_NUM_CHANNELS_2	undefined	lp5xcs2dq18ch2
DWC_LPDDR5XPHY_NUM_CHANNELS_2	defined	lp5x4xcs2dq18ch2

### 8.1.1.1 Customer Defines

Other defines available to customer and provided in example testbench:

- `define DWC\_LPDDR5XPHY\_NUM\_TOP\_SCAN\_CHAINS {s}

DWC\_LPDDR5XPHY\_NUM\_TOP\_SCAN\_CHAINS is provided as the sum of the number of scan chains within the embedded hard IP blocks for a given PHY configuration. This is based on values of 5 DEFINES, as described above. The scan ports are provided as a bundle to the PHY top-level Verilog wrapper, and they can be considered a suggestion dependent on customer balancing and assignment of scan chains in the customer's implementation.

- `define DWC\_LPDDR5XPHY\_CUST\_PHYREV 5'b00000

DWC\_LPDDR5XPHY\_CUST\_PHYREV is provided as a convenience for customers, allowing control of the CUSTPHYREV CSR read-only value. Customers may override this for their application of the PHY. The define only affects the content of the CUSTPHYREV CSR when it is read; it has no other effect on PHY RTL or Firmware.

- `define DWC\_LPDDR5XPHY\_CUST\_PUBREV 5'b00000

DWC\_LPDDR5XPHY\_CUST\_PUBREV is provided as a convenience for customers, allowing control of the CUSTPUBREV CSR read-only value. Customers may override this for their application of the PHY. The define only affects the content of the CUSTPUBREV CSR when it is read; it has no other effect on PHY RTL or Firmware.

- `define DWC\_DDRPHY\_LBIST\_EN

DWC\_DDRPHY\_LBIST\_EN is provided to enable support of optional LogicBist feature in the automotive PHY.

## 8.2 Additional, Optional Customer Defines

### 8.2.1 DWC\_LPDDR5XPHY\_TOP\_PG\_PINS Define

The DWC\_LPDDR5XPHY\_TOP\_PG\_PINS define adds the VDD, VSS, VDDQ, VDD2H ports to the top-level.

In simulation, this define may be provided by customers to drive the logical 0/1 values to VDD, VSS, VDDQ, and VDD2H ports on their PHY instance. It may optionally be used for physical purposes (for example, synthesis) in a flow that requires power and ground pins to be visible as top-level ports. In general, follow the Implementation Guide directions for physical flow configuration.

If the above DWC\_LPDDR5XPHY\_TOP\_PG\_PINS is not defined, the default behavior of the RTL is that a customer uses a upf file/flow to drive top-level power/ground pins during simulation.

### 8.2.2 DWC\_LPDDR5XPHY\_PG\_PINS Define

The DWC\_LPDDR5XPHY\_PG\_PINS define adds the VDD, VSS, VDDQ, VDD2H ports to the hard macro level.

In simulation, this define may be provided by customers to drive the logical 0/1 VDD, VSS, VDDQ, and VDD2H values via ports on hard macro instance. It may optionally be used for physical purposes (for example, synthesis) in a flow that requires power and ground pins to be visible as top-level ports. In general, follow the Integration Guide directions for physical flow configuration.

If the above DWC\_LPDDR5XPHY\_PG\_PINS is not defined, the default behavior of the RTL is that a customer uses a upf file/flow to drive hard macro power/ground pins during simulation.

#### Cautionary Notes on Simulations

The PHY is a highly configurable product with many internally configurable parameters. The following guidance must be followed to ensure proper operation in simulation:

- The "pvtscal" parameters must never be adjusted or changed

### 8.2.3 DWC\_LPDDR5XPHY\_POP\_ENABLED Define

PHY supports both discrete and POP packages. Hard macro instance ordering and PCLK routing must be optimized according to package type. Customer should enable this define if POP package is used.

The DWC\_LPDDR5XPHY\_POP\_ENABLED define selects the PCLK connection topology from the PAC hard macro to repeaters in DX4 hard macros.

By default, DWC\_LPDDR5XPHY\_POP\_ENABLED is un-defined and PCLK repeater connection is optimized for discrete package. Customer should enable DWC\_LPDDR5XPHY\_POP\_ENABLED define to support PCLK repeater connection optimized for PoP package.

### 8.2.4 DWC\_LPDDR5XPHY\_PCLKCA\_RPTR\_ENABLED Define

PHY supports both discrete and POP packages. PHY also supports EW and NS orientations. Hard macro instance ordering and PCLK routing must be optimized according to package type and floorplan orientation.

Customer should enable this define only if NS orientation is used with POP package.

The DWC\_LPDDR5XPHY\_PCLKCA\_RPTR\_ENABLED define selects the PClkCa connection topology from the PAC hard macro to ACX2/CKX2/CSX2 hard macros.

By default, DWC\_LPDDR5XPHY\_PCLKCA\_RPTR\_ENABLED is un-defined and no repeater is used.



# 9

## Design for Test

---

This section provides information regarding the Design For Test features available within the PHY and the associated details for leveraging the individual test capabilities. The PHY supports these tests at nominal Core VDD = 0.75 V only.

The following sections are included in this chapter:

- “Scan Mode” on page [250](#)
- “Loopback BIST Mode” on page [253](#)
- “Loopback BIST Test Strategy and Hardware” on page [254](#)
- “Delay Element Testing” on page [256](#)
- “Analog/Digital Test Observability” on page [258](#)
- “Burn-In (HTOL) Recommendations” on page [277](#)
- “Bypass Mode” on page [278](#)
- “I/O Cell DC Parametric Characterization” on page [286](#)
- “Quiescent Current (IDQ) Measurement” on page [301](#)
- “Input Pin Leakage Measurement” on page [302](#)

## 9.1 Scan Mode

To enter ATPG Test Mode, also known as, Scan Mode, do the following:

1. Power up and Reset the PHY
2. Set BP\_PWROK=1
3. Set scan\_mode = 1

In Scan Mode, following PHY input pins must be driven to High:

- TxBypassMode\_\*
- RxBypassPadEn\_\*
- RxBypassRcvEn\_\*

All possible registers in the PHY blocks are put on scan chains. Each scan chain is based on an internal functional clock group (mixing of clock edges allowed). All scan chains have lockup latches at their outputs. For implementation on PUB, customers should perform scan insertion, stitching and add lockup latches on each chain.

There is a separate scan enable (scan\_shift[\*]) for each clock domain. Scan pins are intentionally left open in the RTL. The scan\_si and scan\_so (scan in/out) are treated as ports at the top-level for all the hard IP instances. The stuck-at scan capture frequency is the same as the stuck-at scan shift frequency. The scan shift frequency is the same in stuck-at or at-speed mode.

Each scan chain in the hard IP macros should have a length of maximum 100 registers. Scan chains belonging to the same group or clocked by the same edge of the clock may be concatenated externally, to build up longer, balanced chains, if required. Since all scan chains use a positive clock edge, it is possible to concatenate all the chains together under the same clock domain, with the maximum chain length in mind.

The following additional features are implemented internally to increase test pattern coverage without additional effort or actions when iddq\_mode=0:

- Reset inputs of infrequently used asynchronous flops are connected to some primary input of the block using a glitch-free bypass-mux.
  - The mux select is controllable from a primary input (scan\_shift\_async or scan\_mode).
- Clock gaters are forced ON during test clock strobing (using scan\_shift\_cg).
- All IO drivers are forced to max drive strengths (using scan\_mode).
- All IO receivers are forced with no ODT strengths (using scan\_mode).

See “[Input Pin Leakage Measurement](#)” on page [302](#) to see how scan\_mode and iddq\_mode interact.



**Note** In ATPG Mode (scan\_mode input pin == 1), all IO drivers are forced to maximum drive strength. Scan is a destructive test thus full PHY reset sequence is required after the completion of the patterns.

To exit Scan Mode, do the following:

1. Start DfiClk
2. Set reset and reset\_async input pin to 1
3. Wait 64 DfiClks
4. Set scan\_mode input pin 0



**Note** PHY supports these tests at nominal PVT (VDD = 0.75 V) corner only.

## 9.1.1 At Speed Scan Test (ASST)

### 9.1.1.1 ASST Overview

PHY supports at-speed scan test (ASST) with on-chip clock (OCC) controllers implemented inside the hard macros to generate the scan clocks for internal PHY functional clocks. The OCC controllers are placed on the scan branch of the clock multiplexers and therefore have no effect on the critical mission-mode clock path.

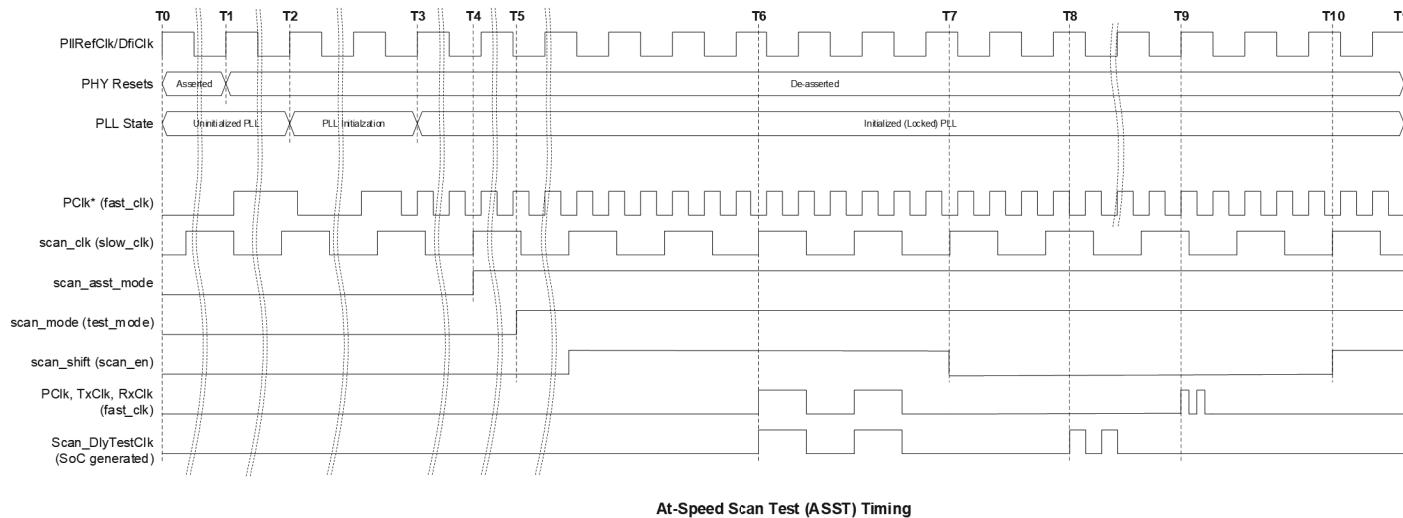
After the PLL is locked, the PHY has uses integrated logic to latch the PLL controls to maintain the PLL state when transitioning into scan testing.

[Figure 9-1](#) shows the sequence for scan testing. The times (Tn) shown in the figure between T0 and T5 are not necessarily consecutive cycles and are used just to show the sequence of operations. The actual timing between operations is dependent on the feature being initialized, such as PLL initialization timing. The timing from T5 are cycle accurate. The following describe the steps for scan sequence:

- T0 to T1: Power up and reset the PHY.  
See “[PHY Initialization](#)” on page [151](#), for details.
- Set csrPwrOkDlyCtrl=1
- Set scan\_occ\_reset to 1 for at least one scan\_clk
- T2 to T3: Startup and lock the PLL by doing the following  
**(PLL in Mission Mode)**
  - Wait a minimum of 1 us
  - Set CPllPwrDn=0
  - Wait a minimum of 1 ns
  - Set CPllPreset=1 and CPllGearShift=1
  - Set CPllCtrl1, CPllCtrl3, and CPllCtrl4 registers according to frequency and Optimal PLL settings in PHY databook.
  - Wait a minimum of 1 us
  - Set CPllPreset=0
  - Wait a minimum of 0.5 us
  - Set CPllGearShift=0

- ❑ Wait a minimum of 10 us for PLL to lock
- (PLL in Bypass Mode)**
- ❑ Wait a minimum of 1 us
  - ❑ Set CPllPwrDn=1
  - ❑ Wait a minimum of 1 ns
  - ❑ Set CPllBypMode=1
  - ❑ Wait a minimum of 1 us
- T4: scan\_asst\_mode to 1. This latches and preserves all the PLL controls to maintain the PLL running at locked state. Wait 16 DFI clock cycles for the latch to take effect.
  - T5: Set scan\_mode to 1 to enter scan test mode and wait 300 ns so that scan\_mode is stable to the final hard macro blocks
  - T6: Set scan\_shift\_async and scan\_shift to 1 to start shifting data in all scan chains, including OCC clock chains.
  - T7: Set scan\_shift\_async and scan\_shift to 0 to capture at speed. Scan capture happens at T9 for RxClk, TxClk and PClk registers, and at T8 for LCDL oscillator clock registers.
  - T10: scan\_shift\_async and scan\_shift to 1 to start shifting data out of scan chains. Scan shift out starts happening at T11.

**Figure 9-1 Scan Testing Sequence**



### 9.1.1.2 PHY\_TOP Scan Clocks

**Table 9-1** **PHY\_TOP Clocking During Scan**

Name	Max Freq	Used by PUB	Used by hard macro	Modes
DfiClk	1200 MHz	Yes	Yes	ALL
APBClk	1200 MHz	Yes	No	ALL
TDRClk	300 MHz	Yes	No	ALL
PllRefClk	DfiClk	No	Yes, PLL only.	ALL
PllBypClk	1600 MHz	No	Yes, PLL only.	ALL
scan_DlyTestClk	1200 MHz	No	Yes (used by RingOsc during scan)	Scan
scan_clk	300 MHz	No	Yes	Scan
scan_UcClk	1200 MHz	Yes (for Arc processor)	No	Scan
Virtual_AsyncPortClk <sup>1</sup>	< 300 MHz			
atpg_Virtual <sup>1</sup>	< 300 MHz			

1. A Virtual clock only for timing purpose.
2. The scan\_clk and TDRClk frequency of 300 MHz is supported only when VDD is 0.75v(nominal).

## 9.2 Loopback BIST Mode

### 9.2.1 Loopback BIST (ATE) Firmware Setup and Execution

The PHY contains features to enable self-contained loopback testing. The loopback hardware is used in conjunction with the PHY ATE Firmware image to enable testing of the IOs and the High speed datapath.

The recommended procedure for loopback is as follows:

- Follow the PHY Initialization procedure to bring up VDD, VDD2H, VDDQ, and VAA ([“PHY Initialization” on page 151](#))
- Follow the PHY initialization procedure to Start Clocks and Reset the PHY ([“PHY Initialization” on page 151](#))
- Load the microcontroller memory with the Loopback BIST firmware
- Lock the DfiClk to the desired Loopback Frequency
- Execute the Loopback BIST Firmware:
  - Initialize the Firmware Message Block with the required test parameters. The content to be written is test dependent, so the documentation provided with each firmware BIST image should be consulted.
  - Enable microcontroller access to the internal CSRs by setting csrMicroContMuxSel to 1. This allows the microcontroller unrestricted access to the configuration CSRs.

- Begin execution of the firmware by setting the MicroReset CSR to 4'b0000.
- Poll mailbox until the microcontroller signals loopback completed.
- Enable external access to the internal CSRs by setting csrMicroContMuxSel to 0. This allows the external APB interface unrestricted access to the configuration CSRs.
- Read the Firmware Message Block to obtain the results from the Loopback BIST.

Data signals are optionally looped back internally or at the I/O pad. If I/O pad loopback is selected, it is important that these points are correctly externally terminated. Data byte DQ/DM and DQS lanes are loop backed to their respective receivers. Loopback data is generated from PPGC (PRBS generator/checker) and checked by DTSM after looping back A PRBS generator and DTSM checkers are implemented for this purpose.

The portion of the Firmware dedicated to Loopback BIST performs the following functions:

- AC and DBYTE PPGC is configured to supply a test pattern. In the case of data loopback, a pre-configured PRBS pattern can be selected as the stimulus.
- Test mode configuration registers are configured for data loopback.
- Tx and Rx delays are set accordingly.
- AC (self-checker) and DBYTE (DTSMs) configured to check for errors.
- DQX1, DQSX2, ACX2, CSX2 and CKX2 slices configured for phase-detect testing. No configuration of phase-detection for ACX2 usage on BP.DTO and CMOS is required.
- ACSM configured to perform a sequence of multiple writes and reads.
- ACSM loopback mode enabled.
- Initiate ACSM run sequence.
- On completion, check the accumulated error counter for each lane.

A typical Address/Command PHY loopback configuration sequence is as follows.



**Note** The following example is for information only. The Loopback BIST Firmware executes the sequence once loaded and activated.

- ACSM configured to perform multiple writes to advance PPGC.
- Initiate ACSM run sequence.
- On completion, check the error counter for each lane.

## 9.3 Loopback BIST Test Strategy and Hardware

The following sections describe how each slice type is tested during loopback.

### 9.3.1 DQX1 Loopback

DQX1 hard macro will be tested by using RxClk\_T/C as the capture clock. It should be tested up to at least the mission-mode frequency. Timings are adjusted via the Tx LCDL using the hard macro microcell input TxDqDly. These may be tested in either core-side loopback (CoreLoopBackMode=1), wherein data is looped back before the pre-driver, or in pad-side loopback (CoreLoopBackMode=0), wherein data is looped back through the mission-mode driver and receiver. When pad-side loopback is used, proper termination at the pad externally must be provided.

- During core-side loopback, SOC can control IO pad state from PHY\_TOP TxBypass pins.
- loopback data is generated from PPGC (PRBS generator/checker) and checked by DTSM after looping back.
- DQX1 should be tested both in RxStrobe mode ( $DxDigStrobeMode[1:0]=2'b0x$ ) and in internally-generated strobe mode to ensure that both paths are functional.
- When internally-generated strobe mode is used during core-side loopback, Tx can be tri-stated and Rx can be power-down to save power.

When internally-generated strobe is used,  $DxDigStrobeMode='b10$  on RxClk LCDLs must be used to sample loopback data.

$TxDqDly$  needs to be swept along with the increment of  $TxDataFIFO$  timing to search for the data and successfully capture PRBS patterns.

- Enabling DFE for pad-side loopback is recommended to ensure DFE control also works.

### 9.3.2 DQSX1 Loopback

In DQSX1, core-side loopback and pad-side loopback take a slightly different paths when looping back the data but both are using the same  $DqsEnArm$ ,  $RxEn$  LCDL's output, to capture data.

- In core-side loopback mode ( $CoreLoopBackMode=1$ ), looping output data ( $TxDataT$  and  $TxDataC$ ) just before the pre-driver back into the input path ( $RxDataCoreT$  and  $RxDataCoreC$  in the diagram). Two built-in phase detectors in the receiver path will check the alignment of the positive-sense output and negative-sense output respectively to the rising edge of  $DqsEnArm$ . A clock pulse will be generated by the PUB on the  $RxStrobeEn$  input to the Hard macro macrocell. The phase detector output is returned to the PUB as  $RxDataPDT$  and  $RxDataPDC$ .
- During core-side loopback, SOC can control IO pad state from PHY\_TOP TxBypass pins.
- In pad-side loopback mode ( $CoreLoopBackMode=0$ ), the output data ( $TxDataT$  and  $TxDataC$ ) will be looped back through the mission-mode driver and receiver as  $RxDataRcvT$  and  $RxDataRcvC$ . In this mode the same phase detectors used to train  $RxEn$  vs read DQS will be re-purposed for loopback testing. These 2 phase detectors, Train and Track PD, are also clocked by  $DqsEnArm$ , so the PUB must generate sampling pulses on it.

The phase detector output is flopped by  $DfiClk$  and returned to the PUB as  $RxDataPD$ . Given the simple flop interface back to the PUB, PUB logic should send a single pulse on  $RxStrobeEn$  to sample loopback data, then wait adequate time to ensure that the phase detector output has stabilized and been cleanly sampled by the  $DfiClk$  flop before observing  $RxDataPD$  within the PUB.

### 9.3.3 ACX1/CSX1 Loopback

Very similar to the implementation of DQX1, this is tested by using  $AcRxClk_T/C$  as the capture clock instead. Two ACX1/CSX1 are built into one ACX2/CSX2 hard macro so there are two pairs of  $AcRxClk_T/C$  in a ACX2/CSX2 hard macro. During core or pad-side loopback, here are the key differences between DQX1 and ACX1/CSX1:

- ACX1/CSX1 can only be tested with internally-generated Pclk by asserting  $csrAcRxClkEn$ .
- During core-side loopback, SOC can control IO pad state from PHY\_TOP TxBypass pins.
- DTO bump is built with AC hard macro and is supported only when  $DWC_LPDDR5XPHY.DTO_ENABLED$  is defined. Only pad-side loopback is supported for such

bump. As DTO is a low speed signal, the loopback data are captured back in DfiPubClk domain and does not require high speed PRBS patterns.

### 9.3.4 CKX2 Loopback

CKX2 loopback is like ACX2/CSX2 (two ACX1/CSX1 together) loopback rather than DQSX1 loopback. They both employ AcRxClk\_T/C enabled by AcRxClkEn instead of RxEn in DQSX1. For this reason, refer to ACX1/CSX1 sampling procedure for loopback operation.

During core-side loopback, SOC can control IO pad state from PHY\_TOP TxBypass pins.

### 9.3.5 CMOSx1 Loopback

CMOSx1 loopback is for loopback testing on MEMRESET\_L pin. It only supports pad level loopback.

The sampling procedure from DQX1 loopback section can be followed as the general guidance except that MEMRESET\_L is a low speed signal, the loopback data are captured back in DfiPubClk domain and does not require high speed PRBS patterns.

### 9.3.6 DTO Loopback

DTO loopback is for loopback testing on BP.DTO pin and is supported only when DWC\_LPDDR5XPHY.DTO\_ENABLED is defined; in addition, it only supports pad level loopback.

Because BP.DTO is built by using AC hard macro, the sampling procedure from ACX1/CSX1 loopback section can be followed as the general guidance except that BP.DTO is a low speed debug signal, the loopback path is through bypass path and is captured back in DfiPubClk domain which does not require high speed PRBS patterns. In order to use the bypass receiver the PAD should not be terminated for the DTO during loopback testing.

## 9.4 Delay Element Testing

### 9.4.1 Overview

- Hard macro Slices have delay element LCDL ("locally calibrated delay line") which are used to achieve fractional UI timing of the DRAM interface signals
  - DQSX1 hard macro has 3 LCDLs: One for Transmit differential signal, one RxStrobeEn Signal (Read Gate) and another for RxReplica
    - For DQSX1 in Dx5 hard macro, its RxReplica LCDL is unused, so no delay element testing is required.
  - DQX1 hard macro has 3 LCDLs: One for Transmit single ended signal and two for Receive Byte Strobe RxStrobeT and RxStrobeC
  - ACX2/CSX2 hard macro has 2 LCDLs: One for each Transmit single ended signal.
  - CKX2 hard macro has 2 LCDLs: One for Transmit differential signal and one for CK primary calibration.
- There is a facility for testing the linearity of delay lines by changing delay setting. This ensures that they will be known to be working correctly for training timing and for locking to the PLL-generated Pclk\_Ca/Pclk\_Dq and tracking Pclk\_Ca/Pclk\_Dq.
- The delay provided by an LCDL consists of two parts:

- ❑ A non-configurable (that is, unavoidable) delay that we call "insertion\_delay" or "zerodelay" and a configurable delay that we call "Phase [8:0]" on the LCDL wrappers.
- ❑ A configurable delay that we call "Phase [8:0]" on the LCDL wrappers.
- ❑ The total delay is given by:
  - $\text{LCDL\_delay} = (\text{Phase}[8:0] * \text{stepsize}) + \text{zerodelay}$ 
    - Where the LCDL time constants zerodelay and stepsize are technology dependent, and vary with instance, process, temperature, and voltage.
    - Maximum Phase[8:0] = 503.
- ❑ A design technique is employed during LCDL calibration to remove the non-configurable delay such that the effective total delay can be:
  - $\text{LCDL\_delay} = (\text{Phase}[8:0] * \text{stepsize})$
- The purpose of the delay test is to check that the individual LCDLs are controllable in this way, with no shorts or opens in the controls to the LCDL and that the LCDL delay circuitry is working.

#### 9.4.2 Delay Test Hardware

- In each Hard macro Slice there is one copy of the delay test hardware corresponding to each LCDL. Each copy of the delay test hardware consists of:
  - ❑ A ring-oscillator, which is a connection from the delay cell output to its input through an inverter
  - ❑ One 16b up-counter clocked by the ring-oscillator.
- The period of the oscillator may be determined from the runtime and the ring-oscillator counts. The period of the oscillator is twice the delay through the delay cell plus a small amount of implementation routing and the inverter delay. Conversely, the expected cell delay, and hence the period, may be converted into an expected count. Running the ring-oscillator for a sequence of phases will provide the information to determine that the LCDLs are controllable and linear as required.
- Set the duration of the interval of single step linearity testing through `csrDlyTestCntDfiClkIVHM`. The longer the test runs the better resolution of the results.
- Assert `csrLcdlCalEn`, then Assert `csrDlyTestEn` and `csrDlyTestEnHM` on the specific hard macro under test, and initialize the counter through `csrDlyTestCntInitHM`.
- Poll the `csrDlyTestCntDfiClkHM` to find the status of testing. The test interval is done when this is zero.
- Programming `csrEnablereadROcntHM` to transfer the count value to DfiClk domain at known stable time.
- `csrDlyTestCntRingOsc` counter value will be available for each Slice and for each LCDL therein.
- Each hard macro will have its individual RO\_Run. Width of RO\_run will be based on number of LCDL in a given hard macro. This is programed through `csrDlyTestMuxSelHM` and `csrDlyTestCntRingOscSel{Dx, ACX}`
  - ❑ ACX2\_TOP: two AC instances and 1 LCDL per AC instance.
    - For BP.DTO which is built from ACX2\_TOP, it is unsupported only when `DWC_LPDDR5XPHY.DTO_ENABLED` is defined. Since only bypass/flyover path is supported for BP.DTO, so no LCDL delay test is needed on this hard macro.
  - ❑ CKX2\_TOP: 2 LCDLs

- Dx4/Dx5\_top:
  - DQSX1: 3 LCDLs
  - DQX1: 3 LCDLs

## 9.5 Analog/Digital Test Observability

### 9.5.1 Digital Observability

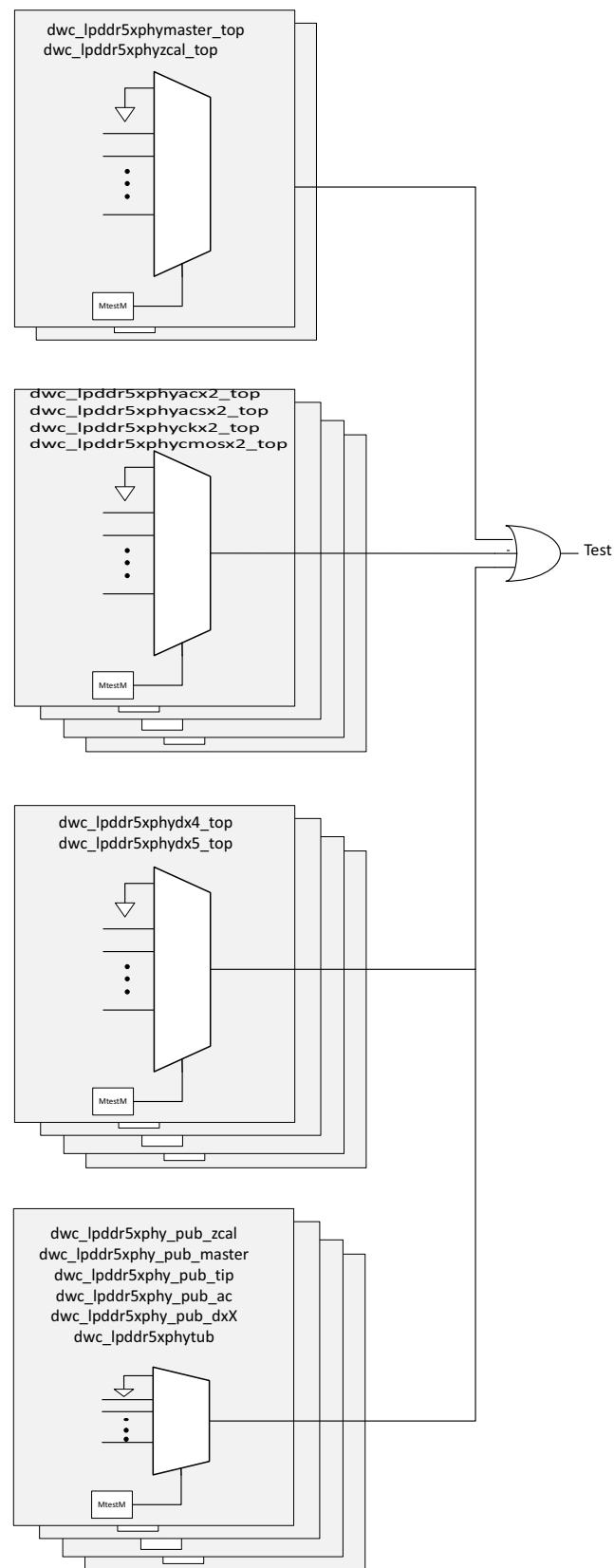
The PHY has the capability to send many of the internal important digital control signals and clock signals (max speed limited to 1066 MHz) to output pins using internal bypass driver on BP.DTO, BP.A[9] or BP.A[19], to aid with test and debug of the PHY. The number of digital test outputs varies based on the configuration of the PHY.



**Note** Bump BP.DTO0 and BP.DTO1 exist only when DWC\_LPDDR5XPHY.DTO\_ENABLED is defined. If DWC\_LPDDR5XPHY.DTO\_ENABLED configuration is not defined, the test output pin will be mapped to BP.A[9] and BP.A[19] (only in dual channel configuration) instead.

The test output selection is built as multiple test selection muxes with all of the mux outputs OR'd together before being sent out to the pins. Each mux has a constant 1'b0 input tied to the 0 input of the mux. The MtestMuxSel CSR controls the selection value to each mux. Setting all the CSR's to 0 will cause all the muxes to select the 1'b0 to the test output, which would cause the test output to be set to a constant 0. Setting one of the mux selection CSRs to a non-zero value will allow that signal value to be sent to the test output. If all other CSR mux selector values are set to 0, then all the other muxes will send 0 which will not interfere with the value of the signal of interest. Therefore, the basic process is:

- Write all the MtestMuxSel CSRs to 0 to force them all to send the constant 0.
- Write the one MtestMuxSel CSR that contains the signal to output to the value to select that signal.

**Figure 9-2 Overall Test Selection Structure**

### 9.5.1.1 Digital Test Observation Pin

The value output on the Digital Test Observation pin is controlled by the setting of the MtestMuxSel CSR. There is a copy of this CSR in each dwc\_lpddr5xphy{\_pub\_dxX, \_pub\_ac, \_pub\_zcal, \_pub\_master, \_pub\_tip, tub, master\_top, zcal\_top, acx2\_top, csx2\_to, cksx2\_top, cmosx2\_top, dx4\_top, dx5\_top} Instance inside the PHY.

Different digital observability pin is used depending on the definition on DWC\_LPDDR5XPHY.DTO\_ENABLED. If DWC\_LPDDR5XPHY.DTO\_ENABLED is undefined, BP\_A[9] pin is used for observability. This is generally being used when in LP5/5x mode. If DWC\_LPDDR5XPHY.DTO\_ENABLED is defined, dedicated BP.DTO pin is used instead and this is used when in LP4x mode. Use the following procedures to conduct the testing:

1. Bring up VDD, VDDQ, VDD2H and VAA (see “[\(A\) Bring up VDD, VDDQ, VDD2H and VAA](#)” on page [153](#))
2. Start Clocks and Reset the PHY (see “[\(B\) Start Clocks and Reset the PHY](#)” on page [153](#))
3. Set PHY pin scan\_mode = 0x0.
4. Set PHY pin iddq\_mode=0x0
5. Perform the following REGISTER write if this it is not done. Skip this If this has been done from other test sequences.
  - a. Write csrPwrOkDlyCtrl = 0x1 to assert power OK at VDD, VDDQ, VDD2H and VAA domain.
  - b. Write csrMTestDtoCtrl =0x1.
6. Perform the following REGISTER write:
  - a. If DWC\_LPDDR5XPHY.DTO\_ENABLED is defined, use BP.DTO for observability:
    - i. Write csrTxStrenCode{PU,PD}AC[\*] with 4'b1111 on AX\_0 of ACx2\_8 instance in AC wrapper 0.
    - ii. Write TxBypassModeExt = 0 and TxBypassOEInt = 1 on AX\_0 of ACx2\_8 instance in AC wrapper 0.
  - b. If DWC\_LPDDR5XPHY.DTO\_ENABLED is undefined, use BP\_A[9] for observability:
    - i. Write csrTxStrenCode{PU,PD}AC[\*] with 4'b1111 on AX\_1 of ACx2\_3 instance in AC wrapper 0.
    - ii. Write TxBypassModeExt = 0 and TxBypassOEInt = 1 on AX\_1 of ACx2\_3 instance in AC wrapper 0.
  - c. Write csrDTOBypassEn=1 of PUB\_AC 0.

7. Set all the selection muxes to select constant zero by writing 0x000 to csrMtestMuxSel in all instances of CSR. Such CSR exists for each hard macro and PUB instance except PAC hard macro.
  - a. Write the MtestMuxSel CSR in a single instance to output the desired signal from a single instance of any one block mentioned above.
    - i. MtestMuxSel[4:0] --> Lower 5 bits selects one of the bit from 32 bit of mux within each Hard-macro(Slice)/PUB instance.
    - ii. MtestMuxSel[8:5] --> Upper 4 bits selects a specific instance of the PHY.
8. For PAC hard macro, csrMtestMuxSelMAS[8:0] is used instead.
  - a. MtestMuxSelMAS[4:0] --> Lower 5 bits selects one of the bit from 32 bit of mux within PAC hard macro.
  - b. MtestMuxSelMAS[8:5] --> set to 4'b0 to select PAC hard macro.



When using BP.DTO, BP.A[9] or BP.A[19] as the observability pin for MtestMux, Async Bypass CSR have no control on PAD.

**Table 9-2 MtestMuxSel PUB or Hard Macro Selection**

PUB or Hard macro	MtestMuxSel[8:5]	Description
dwc_lpddr5xphyzcal_top	0x0	Select ZCAL_TOP instance
dwc_lpddr5xphycmosx2_top	0x0	Select CMOSX2_TOP instance
AX_0 of dwc_lpddr5xphyacx2_top CS_0 of dwc_lpddr5xphycsx2_top	Mux-A: 0x0, Mux-B: 0x1	Program CSR from one of ACX2_TOP/CSX2_TOP instances that needs to be tested and choose one of two Muxs within each instance.
AX_1 of dwc_lpddr5xphyacx2_top CS_1 of dwc_lpddr5xphycsx2_top	Mux-A: 0x2, Mux-B: 0x3	
dwc_lpddr5xphyckx2_top	Mux-A: 0x0, Mux-B: 0x1	Program CSR from CKX2_TOP instances and choose one of two Muxs within each instance.
dwc_lpddr5xphydx4_top	Mux-A : 0x{0...3} for DQ{0...3} ; 0x5 for DQS Mux-B : 0x{8...B} for DQ{0...3} ; 0xD for DQS	Program CSR for one of Dx4_top instances, within each instance and choose one of two Muxs for each DQ/DQS.
dwc_lpddr5xphydx5_top	Mux-A : 0x{0...4} for DQ{0...4} ; 0x5 for DQS Mux-B : 0x{8...C} for DQ{0...4} ; 0xD for DQS	Program CSR for one of Dx5_top instances, within each instance and choose one of two Muxs for each DQ/DQS.
dwc_lpddr5xphy_pub_zcal	0x0	Select PUB_ZCAL instance
dwc_lpddr5xphy_pub_master	0x0	Select PUB_PAC instance
dwc_lpddr5xphy_pub_ac	0x00	Select PUB_AC instance
dwc_lpddr5xphy_pub_dxX	Mux-A : 0x0, Mux-B : 0x1, Mux-C : 0x2	Select PUB_DxX instances and choose one of three Muxs within each instance.
dwc_lpddr5xphy_pub_tip	0x0	Select PUB_TIP instance
dwc_lpddr5xphy_tub	0x0	Select TUB instance

Only a single instance of a MtestMuxSel CSR can be non-zero at a time. To select a different value, repeat steps 7 and 8. The value to write to the MtestMuxSel CSR in Step 2 is based on the instance that is being written. The following tables describe the various signals that are available for output to the Primary Digital Observation Pin by programming MtestMuxSel[4:0] of each instance.



**Note** Any value not enumerated in the tables below is reserved and should not be programmed

**Table 9-3 Primary Digital Observation Pin Signals Available in PUB ZCAL Instance**

<b>dwc_lpddr5xphy_pub_zcal</b>	
<b>MtestMuxSel[4:0]</b>	<b>Description</b>
31	1'b0
30::28	Mtest_ZCal_stJumpSearch
27	Mtest_CodeFound
26	ZCalCompEn
25	Mtest_ZCal_CalUpdate_code
24	CalUpdate
23	ZCAL_wr_hit_C_Acfg
22	ZCAL_rd_hit_C_Acfg
21	ZCalPDEn
20	ZCalPUEn
19	Mtest_ZCal_SearchSeqDone
18	Mtest_ZCal_TestResultIn
17	Mtest_ZCal_SearchSeqKickOff
16:12	Mtest_ZCal_stZCal[4:0]
11	Mtest_ZCal_ZCalAnaEn
10::8	Mtest_ZCal_stZCont[2:0]
7	Mtest_ZCal_uSTimerEq0
6	Mtest_ZCal_mSTimerEq0
5	Mtest_ZCal_ZCalTrigger
4	PUB_HMZCAL_DfiGaterClk
3	ZCalFirstRunDone
2	Mtest_ZCal_ZCalMeasDone
1	Mtest_ZCal_ZCalBusy
0	1'b0

**Table 9-4 Primary Digital Observation Pin Signals Available in PUB\_PAC Instance**

dwc_lpddr5xphy_pub_master	
MtestMuxSel[4:0]	Description
31	Dfi1Enable
30	Dfi0Enable
29	MTESTDtoEn
28	csrNeverGateCsrClock
27	EnRxDqsTracking
26	EnDqsSampNegRxEn
25	DqsSampNegRxEnSense
24	TrainUpdatePhaseOnLongBubble
23::20	4'd0
19:16	{3'd0, csrPipeNetDis}
15:12	{3'd0, csrDxOutPipeEn}
11:08	{3'd0, csrDxInPipeEn}
7:04	{3'd0, csrAclnPipeEn}
3	csrMtestPgmlInfo
2	csrLP5Mode
1	1'b0
0	1'b0

**Table 9-5 Primary Digital Observation Pin Signals Available in PUB\_TIP Instance**

dwc_lpddr5xphy_pub_tip	
MtestMuxSel[4:0]	Description
31	arc_rdy
30	csrWaitCondAPB
29	csrWaitCondUC
28	z_flag_mtest
27	execute_instr_mtest
26	clear_pipe_mtest
25	advance_pipe_mtest

<b>dwc_lpddr5xphy_pub_tip</b>	
<b>MtestMuxSel[4:0]</b>	<b>Description</b>
24	halt_pie_mtest
23	waiting_mtest
22	awaiting_read_data_mtest
21	take_branch_mtest
20	wait_condition_done_mtest
19	PieClkGaterEnable_mtest
18	terminate_mtest
17	seq0B_CfgCmdField[1]
16	seq0B_CfgCmdField[0]
15	SkipMrw
14	pie_read
13	seq_is_running
12	PIEstackOverflow
11	PIEstackUnderflow
10	PIEOverlappingRead
9	pmi_active
8	dfilInitStart_merge
7	acsmDone
6	PiILock
5	ddrPMReq_merge
4	PIE_Rd_Complete
3	UC_CfgDone_P
2	PIEParityErr
1	PIEProgErr
0	1'b0

**Table 9-6 Primary Digital Observation Pin Signals Available in Each PUB AC Instance**

<b>dwc_lpddr5xphy_pub_ac</b>	
<b>MtestMuxSel[4:0]</b>	<b>Description</b>
31::24	8'd0
23	SwitchHMSlave_CK0
22	BypassActive_CK0
21	CK_StopClk
20::19	2'd0
18	OnetoTwoMode
17	LpCaCtrlStopClk_in1
16	AC_DIIStopArm_R
15	UpdatePhase1UI_CK
14	UpdatePhase1UI
13	LcdlCalSelPhaseX_CK
12	LcdlCalSelPhaseX
11	LcdlCalActive_CK
10	LcdlCalActive
9	Lcdl_stop
8	LcdlCalResetRelock
7	LcdlDly_max_R
6	LcdlInit
5	LpCtrlEn
4	PtrlInitPipe0
3	1'b0
2	dfi_dram_clk_disable_ac
1	internal_dfi_init
0	1'b0

**Table 9-7 Primary Digital Observation Pin Signals Available in Each PUB Dx Instance for Mux A**

<b>dwc_lpddr5xphy_pub_dxX for Mux A</b>	
<b>MtestMuxSel[4:0]</b>	<b>Description</b>
31	seq_TxOErst[3]
30	seq_TxOErst[2]
29	seq_TxOErst[1]
28	seq_TxOErst[0]
27	LcdlCalActive
26	LcdlCalPending
25	CmdFifoInitBusy
24	Init_RstLcdlCal
23	RxPipeInit
22	DoUpdate
21	pushRxTg
20	DqsSampNegRxEnVal
19	Wck2CkTrain
18	LcdlCalSelPhase
17	WrLevTrain
16	RxEnTrain
15	WDQSEXTENSION
14	TrainEnRxEn
13	UIMode
12	csrLP5Mode
11	WCKEXTENSION
10	BypassForceDQ_ptrinit
9	PipeInit
8	PulseDbyteDIIUpdatePhase
7	PState_IN[1]
6	PState_IN[0]
5	RxDATArcvPDT_DQS

<b>dwc_lpddr5xphy_pub_dxX for Mux A</b>	
<b>MtestMuxSel[4:0]</b>	<b>Description</b>
4	1'b0
3	DfiClk
2	1'b0
1	HwtMemSrc_IN_dst[3]
0	1'b0

**Table 9-8 Primary Digital Observation Pin Signals Available in Each PUB Dx Instance for Mux B**

<b>dwc_lpddr5xphy_pub_dxX for Mux B</b>	
<b>MtestMuxSel[4:0]</b>	<b>Description</b>
31	1'b0
30	rxpipe_RxPtr_push
29	MRL_OUT
28	RDFptrA_valid_Ln0_q
27	RdfPtrChkErr
26	i_aux8_comb
25	i_aux4_comb
24	i_aux2_comb
23	i_aux1_comb
22:12	Mtest_rxppt[10:0]
11:01	Mtest_txppt[10:0]
0	1'b0

**Table 9-9 Primary Digital Observation Pin Signals Available in Each PUB Dx Instance for Mux C**

<b>dwc_lpddr5xphy_pub_dxX for Mux C</b>	
<b>MtestMuxSel[4:0]</b>	<b>Description</b>
31::28	RxReplica_LcdlBypMode
27::24	RxReplica_LcdlReset
23	RxRepLcdlCalActive
22	1'b0

<b>dwc_lpddr5xphy_pub_dxX for Mux C</b>	
<b>MtestMuxSel[4:0]</b>	<b>Description</b>
21	RxReplica_CalPhaseUpdate
20	RxRep_spare5[0]
19:16	RxReplica_LcdlCalClkEn
15:12	RxReplica_LcdlCalPhaseUpdate
11:08	RxReplica_LcdlCalSeqEn
7:04	RxReplica_LcdlMode
3	phui_repCalSeqEn
2	phui_repCalClkEn
1	phui_repCalReset
0	1'b0

**Table 9-10 Primary Digital Observation Pin Signals Available in TUB Instance**

<b>dwc_lpddr5xphytub</b>	
<b>MtestMuxSel[4:0]</b>	<b>Description</b>
31::30	pmi_hold_update_ctr
29::28	lp_ctrl_wakeup_pipe_1_ge_2
27::26	hwt_low_speed_clk_ptrn
25	SkipMrw
24	pm_req_done
23	PhyAlert
22	HwtMemSrc0_r[1]
21	arc_rdy
20	ddrPMReq
19	hclkEn
18:17	LongBubble
16	hwt_low_speed_clk_enbl
15	pmi_ddrPMReq
14:10	pmi_dfiFreq
9:08	pmi_lp_in_progress_c

dwc_lpddr5xphytub	
MtestMuxSel[4:0]	Description
7:06	LpCtrlEn
5	hwt_phyupd_req_c0
4	PtrlInitPipe0_0
3	pmi_active
2	UcPhasor
1	APB_CfgRdHit
0	1'b0

### 9.5.1.2 Using Two Observation Pins to Assist PLL Debug

PLL provides two-bit digital test outputs to assist PLL debug. These two bits can be observed through the BP\_DTO pin by configuring the MtestMuxSel CSR, specifically on MtestMuxSel[15:14] of dwc\_lpddr5xphymaster\_top covered in one of the tables above. However, sometimes it would be helpful to observe these two bits at the same time, so in addition to the BP\_DTO pin, a second pin is also being provided to facilitate this feature. The table below shows the observability from PLL digital test outputs:

**Table 9-11 PLL Digital Test Decoding vs PLL Digital Test Outputs when PLL Bypass = 0**

csrCPIIDigTstSel12:7]	PIIDigTst[1]	PIIDigTst[0]	Description
000 - default	0	0	This code should be set during normal use to minimize clock jitter
1	fbk_clk_ldet	pllin_x1	Buffered version of fbk_clk at PFD input (fbk_clk_ldet freq=pllin_x1/input_divider_ratio)
10	ref_clk_ldet	pllin_x1	Buffered version of ref_clk at PFD input (ref_clk_ldet freq=pllin_x1/input_divider_ratio)
11	pllin_x1	pllin_x1	For calibrating delay skew between two test outputs - input reference clock
100	clk_fbk	pllin_x1*	Buffered version of plfout_x1 (clk_fbk freq=pllin_x1)
101	plfout_clk	pllin_x1*	PLL clock output (Fout<Fmax(dto only)) (plfout_amux_clk freq = 4*pllin_x1 or 2*pllin_x1)
110	en_count	pll_lock	Lock detector state test
111	standby_eff	Eoc	Calibration state test

**Table 9-12 PLL Digital Test Decoding vs PLL Digital Test Outputs when PLL Bypass = 1**

csrCPIIDigTstSel12:7]	PIIDigTst[1]	PIIDigTst[0]	Description
000 - default	0	0	This code should be set during normal use to minimize clock jitter
001	fbk_clk_ldet	pllin_x1	Buffered version of fbk_clk at PFD input (fbk_clk_ldet freq=pllin_x1/input_divider_ratio)
010	ref_clk_ldet	pllin_x1	Buffered version of ref_clk at PFD input (ref_clk_ldet freq=pllin_x1/input_divider_ratio)
011	pllin_x1	pllin_x1	For calibrating delay skew between two test outputs - input reference clock
100	clk_fbk	pllin_x1*	Buffered version of pllout_x1 (clk_fb freq=pllin_x1)
101	pllout_clk	pllin_x1*	PLL clock output (Fout<Fmax_dto only) (pllout_amux_clk freq = bypass_pllin_x1)
110	en_count	pll_lock	Lock detector state test
111	standby_eff	eoc	Calibration state test

- Notice only 3 out of 6 bits of csrCPIIDigTstSel are used. Three MSBs must be 0.
- When two PLL test outputs are needed simultaneously, PLL digital test outputs are mapped to two pins as the following, see [Table 9-13](#).

**Table 9-13 PLL Digital Test Outputs to Bump Mapping**

	PIIDigTst[0]		PIIDigTst[1]	
	When DWC_LPDDR5XPHY_D TO_ENABLED is defined	When DWC_LPDDR5XPHY_D TO_ENABLED is not defined	When DWC_LPDDR5XPHY_D TO_ENABLED is defined	When DWC_LPDDR5XPHY_DT O_ENABLED is not defined
RTL Bump Name	BP.DTO	BP.A[9]	BP.DTO	BP.A[19]

To output the PLL test signals on these two pins, use the following procedure:

- Setup: Power on and reset the PHY. The digital observation pins can be enabled at any point after the reset.
- Step 1: Enable the test outputs.
  - PIIDigTst[0]: depending on the 'define' as listed in the table above, either BP.DTO or BP.A[9] pin is used as output. Follow the description at "[Digital Test Observation Pin](#)" on page 260 for configuring the MtestMuxSel CSR and choosing PIIDigTst[0] as the output.

- Step 2: Use the CPlIDigTstSel CSR in dwc\_lpddr5xphy\_pub\_master to select the value. See [Table 9-13](#) on page [272](#), for the possible values to use.

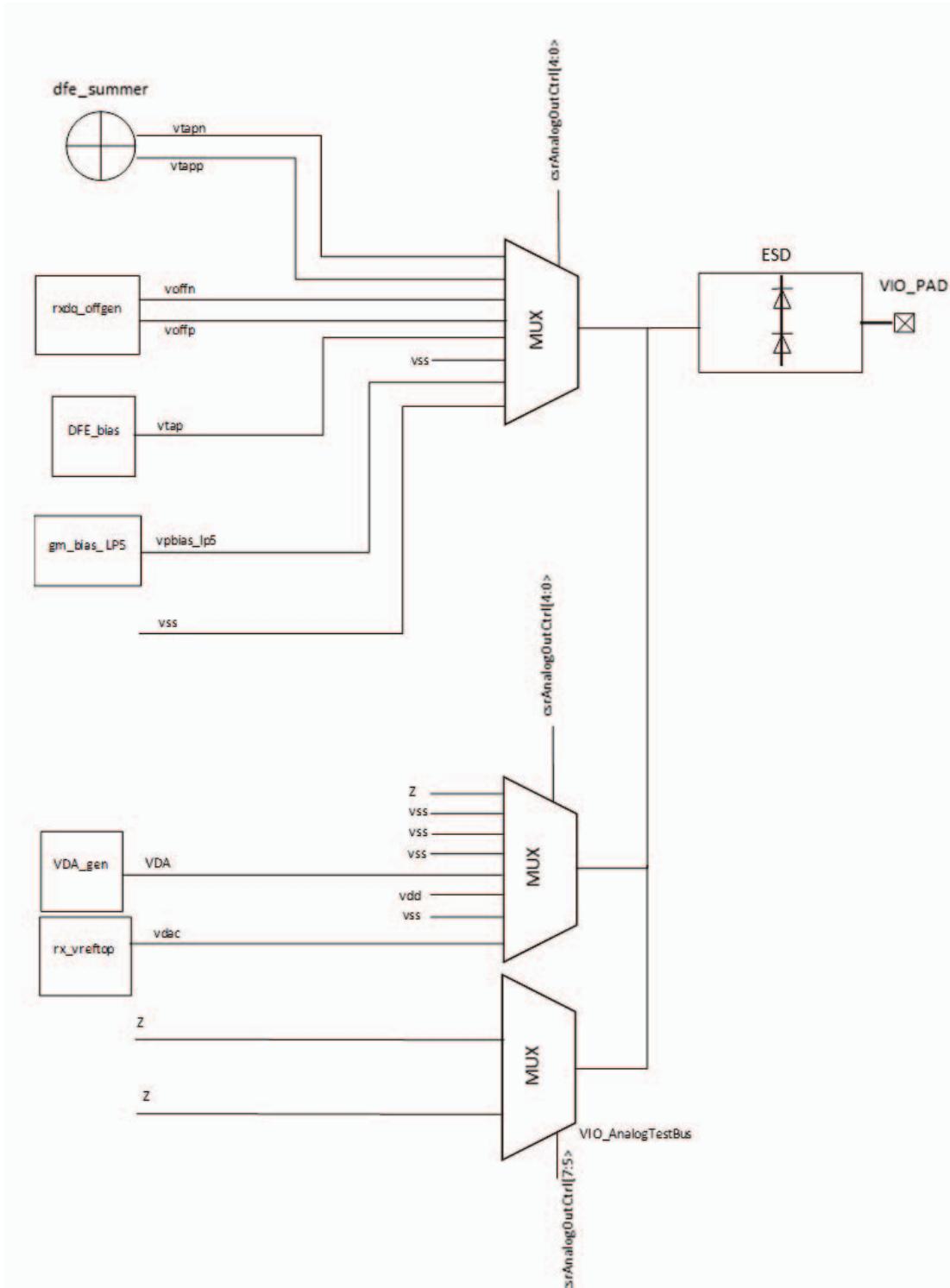
### 9.5.2 Analog Observability

The PHY has the capability to send out analog signal to BP\_ATO and BP\_ATO\_PLL output pins, to aid with test and debug of the PHY. Due to different potentials on these two pins, they should not be shorted in any way.

Notice that BP\_ATO's state could be X in S3 mode for LPDDR5 operation, so this pin should be unconnected during such state.

To observe correct voltage levels on BP\_ATO, it is required to write csrPwrOkDlyCtrl = 0x1.

BP\_ATO is connected to observe the analog behavior of the following blocks in ZCAL hard macro:

**Figure 9-3 BP\_ATO in ZCAL Hard Macro**

The following table shows the selection of a specific function out of this BP\_ATO pins:

**Table 9-14 BP\_ATO Function Selection**

VIO_PwrOK <sup>a</sup>	scan_mode	IDDQ	ATO Function	CsrAnalogOut Ctrl[7:5]	CsrAnalogOut Ctrl[4:0]	IOPAD
0	X	X	N/A	X	X	Z
1	1	X	Z	X	X	Z
1	X	1	Z	X	X	Z
1	0	0	Z	0	0	Z
1	0	0	ATO supplies	0	1	VSS
1	0	0	ATO supplies	0	10	VSS
1	0	0	ATO supplies	0	11	VSS
1	0	0	ATO TX supplies	0	100	VSS
1	0	0	ATO supplies	0	101	VDD
1	0	0	ATO supplies	0	110	VSS
1	0	0	ATO RX DQ	0	111	VDAC <sup>b</sup>
1	0	0	ATO RX DQ	0	1000	vtapn
1	0	0	ATO RX DQ	0	1001	vtapp
1	0	0	ATO RX DQ	0	1010	voffn
1	0	0	ATO RX DQ	0	1011	voffp
1	0	0	ATO RX DQ	0	1100	vtap
1	0	0	ATO RX DQ	0	1101	VSS
1	0	0	ATO RX DQ	0	1110	Vpbias_lp5
1	0	0	ATO RX DQ	0	1111	VSS
1	0	0	Reserved	0	1----	Z (Reserved)
1	0	0	Reserved	1	----	Z (Reserved)
1	0	0	Reserved	10	----	Z (Reserved)
1	0	0	Reserved	11	----	Z (Reserved)
1	0	0	Reserved	1--	----	Z (Reserved)

a. VIO\_PwrOK = BP\_PWROK when csrPwrOkDlyCtrl =1 .

b. VDAC voltage is set by csrRxVrefDac.

BP\_ATO\_PLL is connected to monitor pll\_analog\_test (PLL analog test output pin). To control this feature, use the CSRs below:

- PllAnaTstEn - enable the PLL analog test port.
- PllAnaTstSel - select which PLL signal to observe. Must be equal to 6'b00\_0000 during mission mode.

**Table 9-15 BP\_ATO\_PLL Function Selection**

VIO_PwrOk	PllAnaTstEn	BP_ATO_PLL
0	X	Hi-Z
1	0	Hi-Z
1	1	Depends on CPllAnaTstSel CSR

For CPllAnaTstSel decoder value, refer to PHY databook.

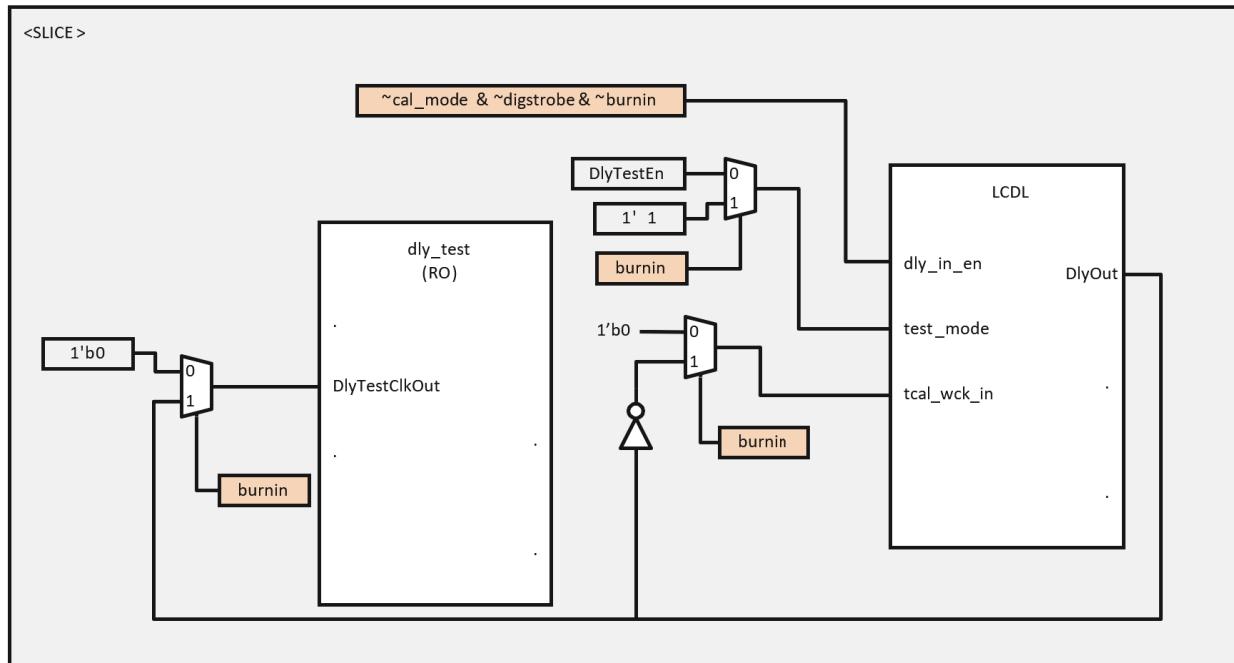
## 9.6 Burn-In (HTOL) Recommendations

During burn-in (HTOL) testing, the recommended action is to run a combination of “Loopback BIST Mode” on page 253 and “Scan Mode” on page 250. This will ensure maximum toggling of internal nodes.

The recommendation is to alternate equally between the two modes throughout the burn-in testing process:

- For 50% of the duration, activate Loopback BIST Mode
  - Loopback BIST Mode is enabled through loading and running of the Loopback BIST Firmware Image.
  - It is recommended to run PLL at the highest mission mode frequency for at speed test coverage. Other frequency or PLL bypass mode is also allowed depending on customers test consideration
  - For core-side and pad-side loopback, it is important that PADs are correctly externally terminated.
- For 50% of the duration, activate SCAN Test Mode
  - Refer to “Scan Mode” on page 250 to bring up the procedure if not been done already.
  - Operate in Scan Shift Mode.
  - Activate all Scan Chains simultaneously.
  - PLL speed is irrelevant as PLL is placed in bypass mode.
  - Drive “BurnIn” pin = 1, it conditions all Slices in Dbyte and AC:
    - Forces LCDL delay select to max value.
    - Forces LCDL into a feedback loop of its own.
- Connection between Burn-In, LCDL Element and Ring Oscillator (Delay Test), see [Figure 9-4](#).

**Figure 9-4** Burn-In Control of LCDL



## 9.7 Bypass Mode

The Bypass interface is provided to enable fully independent control of the PHY I/Os from customer implemented logic external to the PHY.

At PHY top, when the TxBypassMode\* pins are asserted, the associated PHY I/Os controlling BP\_A[\*], BP\_B{\*}\_D[\*], BP\_CK\*[\*], BP.DTO, and BP\_MEMRESET\_L are placed into asynchronous Tx Bypass Mode, and are directly controlled by the top level inputs TxBypass\*. When used in Tx Bypass Mode, the PHY I/O terminations are disabled.

At PHY top, when the RxBypass\*En\* pins are asserted, the RX bypass path is enabled for the associated PHY I/Os controlling BP\_A[\*], BP\_B{\*}\_D[\*], BP\_CK\*[\*], BP.DTO and BP\_MEMRESET\_L ).

Some possible use cases for the Bypass Mode of the PHY I/Os are: Boundary Scan, DC IO Parametric Testing, SoC level ATPG IO, Memory Connectivity Testing. Note that these use cases will require external customer-implemented logic/equipment.

**Table 9-16** shows the Bypass mapping from the PHY Top Bypass Signal to the Bumps. Boundary scan is added by the customer by connecting the various Bypass signals (TxBypassMode\*, RxBypassPadEn\*, RxBypassRcvEn\*, TxBypassOE\*, TxBypassData\*, RxBypassData\*) , (BP\_A[\*],BP\_B{\*}\_D[\*], BP\_CK\*[\*], DTO, MEMRESET\_L ) in the customer wrapper code to create the desired boundary-scan order and behavior. The PHY architecture/RTL is fully flexible with respect to this ordering.

**Table 9-16 Bypass Mapping from the Bypass Signal to the Bumps (Single Channel)**

Bypass Signal	Bumps
TxBypass{Data,OE,Mode}_A[9:0]	BP_A[9:0]
RxBypass{DataPad,PadEn,RcvEn}_A[9:0]	
RxBypassDataRcv_A{9:0}[1:0]	
TxBypass{Data,OE,Mode}_B{NUM_DBYTES-1:0}_D{7:0}	BP_B{NUM_DBYTES-1:0}_D{7:0}
RxBypass{DataPad,PadEn,RcvEn}_B{NUM_DBYTES-1:0}_D{7:0}	(for DQ bits)
RxBypassDataRcv_B{NUM_DBYTES-1:0}_D{7:0}[1:0]	
TxBypass{Data,OE,Mode}_B{NUM_DBYTES-1:0}_D{12}	BP_B{NUM_DBYTES-1:0}_D{12}
RxBypass{DataPad,PadEn,RcvEn}_B{NUM_DBYTES-1:0}_D{12}	(for DMI/DM bits)
RxBypassDataRcv_B{NUM_DBYTES-1:0}_D{12}[1:0]	
TxBypass{Data,OE,Mode}_B{NUM_DBYTES-1:0}_D{9:8}	BP_B{NUM_DBYTES-1:0}_D{9:8}
RxBypass{DataPad,PadEn,RcvEn}_B{NUM_DBYTES-1:0}_D{9:8} <sup>a</sup>	(for DQS bits)
RxBypassDataRcv_B{NUM_DBYTES-1:0}_D{9:8}[1:0] <sup>a</sup>	
TxBypass{Data,OE,Mode}_B{NUM_DBYTES-1:0}_D{11:10}	BP_B{NUM_DBYTES-1:0}_D{11:10}
RxBypass{DataPad,PadEn,RcvEn}_B{NUM_DBYTES-1:0}_D{11:10} <sup>a</sup>	(for WCK bits, valid in LPDDR5/5X mode only)
RxBypassDataRcv_B{NUM_DBYTES-1:0}_D{11:10}[1:0] <sup>a</sup>	
TxBypass{Data,OE}_CK0_{T,C}	BP_CK0_T, BP_CK0_C
TxBypassMode_CK0_{T,C}	
RxBypassDataPad_CK0_{T,C}	
RxBypass{Pad,Rcv}En_CK0	
RxBypassDataRcv_CK0_{T,C}[1:0]	

Bypass Signal	Bumps
TxBypass{Data,OE,Mode}_DTO0 RxBypass{DataPad,PadEn,RcvEn}_DTO0 RxBypassDataRcv.DTO0[1:0]	BP.DTO0 This bump will not exist when DWC_LPDDR5XPHY.DTO_ENABLED define is not selected.
TxBypass{Data,OE,Mode}_DTO1 RxBypass{DataPad,PadEn,RcvEn}_DTO1 RxBypassDataRcv.DTO1[1:0]	BP.DTO1 This bump will not exist when DWC_LPDDR5XPHY.DTO_ENABLED define is not selected.
TxBypass{Data,Mode}_MEMRESET_L RxBypassDataPad_MEMRESET_L RxBypassPadEn_MEMRESET_L	BP.MEMRESET.L

- a. The following ports are unused:  
 - RxBypass{PadEn, RcvEn}\_B{NUM\_DBYTES:0}\_D{10:8}  
 - RxBypassDataRcv\_B{NUM\_DBYTES:0}\_D{11:8}[1]

**Table 9-17 Bypass Mapping from the Bypass Signal to the Bumps (Dual Channel)**

Bypass Signal	Bumps
TxBypass{Data,OE,Mode}_A[19:0] RxBypass{DataPad,PadEn, RcvEn}_A[19:0] RxBypassDataRcv_A[19:0][1:0]	BP.A[19:0]
TxBypass{Data,OE,Mode}_B{NUM_DBYTES-1:0}_D{7:0} RxBypass{DataPad,PadEn,RcvEn}_B{NUM_DBYTES-1:0}_D{7:0} RxBypassDataRcv_B{NUM_DBYTES-1:0}_D{7:0}[1:0]	BP.B{NUM_DBYTES-1:0}_D[7:0] (for DQ bits)
TxBypass{Data,OE,Mode}_B{NUM_DBYTES-1:0}_D{12} RxBypass{DataPad,PadEn,RcvEn}_B{NUM_DBYTES-1:0}_D{12} RxBypassDataRcv_B{NUM_DBYTES-1:0}_D{12}[1:0]	BP.B{NUM_DBYTES-1:0}_D[12] (for DMI/DM bits)
TxBypass{Data,OE,Mode}_B{NUM_DBYTES-1:0}_D{9:8} RxBypass{DataPad,PadEn,RcvEn}_B{NUM_DBYTES-1:0}_D{9:8} <sup>a</sup> RxBypassDataRcv_B{NUM_DBYTES-1:0}_D{9:8}[1:0] <sup>a</sup>	BP.B{NUM_DBYTES-1:0}_D[9:8] (for DQS bits)
TxBypass{Data,OE,Mode}_B{NUM_DBYTES-1:0}_D{11:10} RxBypass{DataPad,PadEn,RcvEn}_B{NUM_DBYTES-1:0}_D{11:10} <sup>a</sup> RxBypassDataRcv_B{NUM_DBYTES-1:0}_D{11:10}[1:0] <sup>a</sup>	BP.B{NUM_DBYTES-1:0}_D[11:10] (for WCK bits)
TxBypass{Data,OE}_CK{1:0}_{T,C} TxBypassMode_CK{1:0}_{T,C} RxBypassDataPad_CK{1:0}_{T,C} RxBypassDataRcv_CK{1:0}_{T,C}[1:0] RxBypass{Pad,Rcv}En_CK{1:0}	BP.CK{1:0}_T, BP.CK{1:0}_C

Bypass Signal	Bumps
TxBypass{Data,OE,Mode}_DTO0 RxBypass{DataPad,PadEn,RcvEn}_DTO0 RxBypassDataRcv.DTO0[1:0]	BP.DTO0 This bump will not exist when DWC_LPDDR5XPHY.DTO_ENABLED define is not selected.
TxBypass{Data,OE,Mode}_DTO1 RxBypass{DataPad,PadEn,RcvEn}_DTO1 RxBypassDataRcv.DTO1[1:0]	BP.DTO1 This bump will not exist when DWC_LPDDR5XPHY.DTO_ENABLED define is not selected.
TxBypass{Data,Mode}_MEMRESET_L RxBypassDataPad_MEMORYRESET_L RxBypassPadEn_MEMORYRESET_L	BP_MEMORYRESET_L

- a. The following ports are unused:
- RxBypass{PadEn, RcvEn}\_B{NUM\_DBYTES:0}\_D{10:8}
  - RxBypassDataRcv\_B{NUM\_DBYTES:0}\_D{11:8}[1]

To meet the IEEE 1149.1 Boundary Scan requirements, the asynchronous mission mode and bypass receivers can be captured into the IEEE 1149.1 Boundary Scan Registers simultaneously. The following additional logic and controls needs to be considered to support the IEEE 1149.1 instructions for DDR.

Boundary Scan control logic on both the RxBypassDataPadEn\* and RxBypassRcvEn\* input pins.

Boundary Scan register logic on both the RxBypassDataPad\_\* and RxBypassDataRcv\* data output pins.

Initialization of the PHY and ability to control the IO related CSR settings via TDR interface during the Boundary Scan setup phase.

This is required since the mission mode receivers are not enabled by default.



**Note** Verification of the IEEE 1149.1-2013 "Component initialization instructions and procedures" is currently limited by the TMAX tool BSD support (version 2022.12-SP7).

### 9.7.1 Bypass Mode Rules

The following rules apply to the Bypass Mode control signals:

- Most Bypass signals must be Low in mission mode. Only RxBypassRcvEn\*/ RxBypassPadEn\* may be enabled during mission mode.
- When RxBypassRcvEn\*/ RxBypassPadEn\* is High, IOs must not be in PowerDown or StandBy state.
- TxBypassMode\* and RxBypassPadEn\* must be High for scan testing.  
In LPDDR5/5X mode, RxBypassPadEn=1 is supported only when VDDQ=0.5v .
- PHY input iddq\_mode must be 0.
- Generally BumpData = TxBypassMode && TxBypassOE && TxBypassData.

- Bypass signals on BP.DTO\* is supported only when DWC\_LPDDR5XPHY.DTO\_ENABLED is defined, and only TxBypass{Data,OE,Mode}\_DTO\* and RxBypass{DataPad,PadEn}\_DTO\* signals are supported.

RxBypassRcvEn.DTO\* should be tied to 0 and RxBypassDataRcv.DTO\* output is unused as a result. This is applicable to Boundary Scan and VIH/VIL test as well.

Due to the design of BP\_MEMRESET\_L, the following rules apply to that I/O:

- BP\_MEMRESET\_L does not support tristate in Bypass Mode
  - This pin does not accept the JTAG HIGHZ command.
- BP\_MEMRESET\_L is output-only pad

BypassMode control of the PHY I/Os is available for use after the full power-on and reset sequence of the PHY is complete (see “[PHY Initialization Sequence](#)” on page [151](#), steps A, B and C).

For applications where BypassMode control is required prior to the completion of the PHY reset sequence, the primary PHY input, scan\_mode, can be used with BypassMode controls. Notice iddq\_mode has to be disable in this case. Refer to the following truth table to enable the functionality.

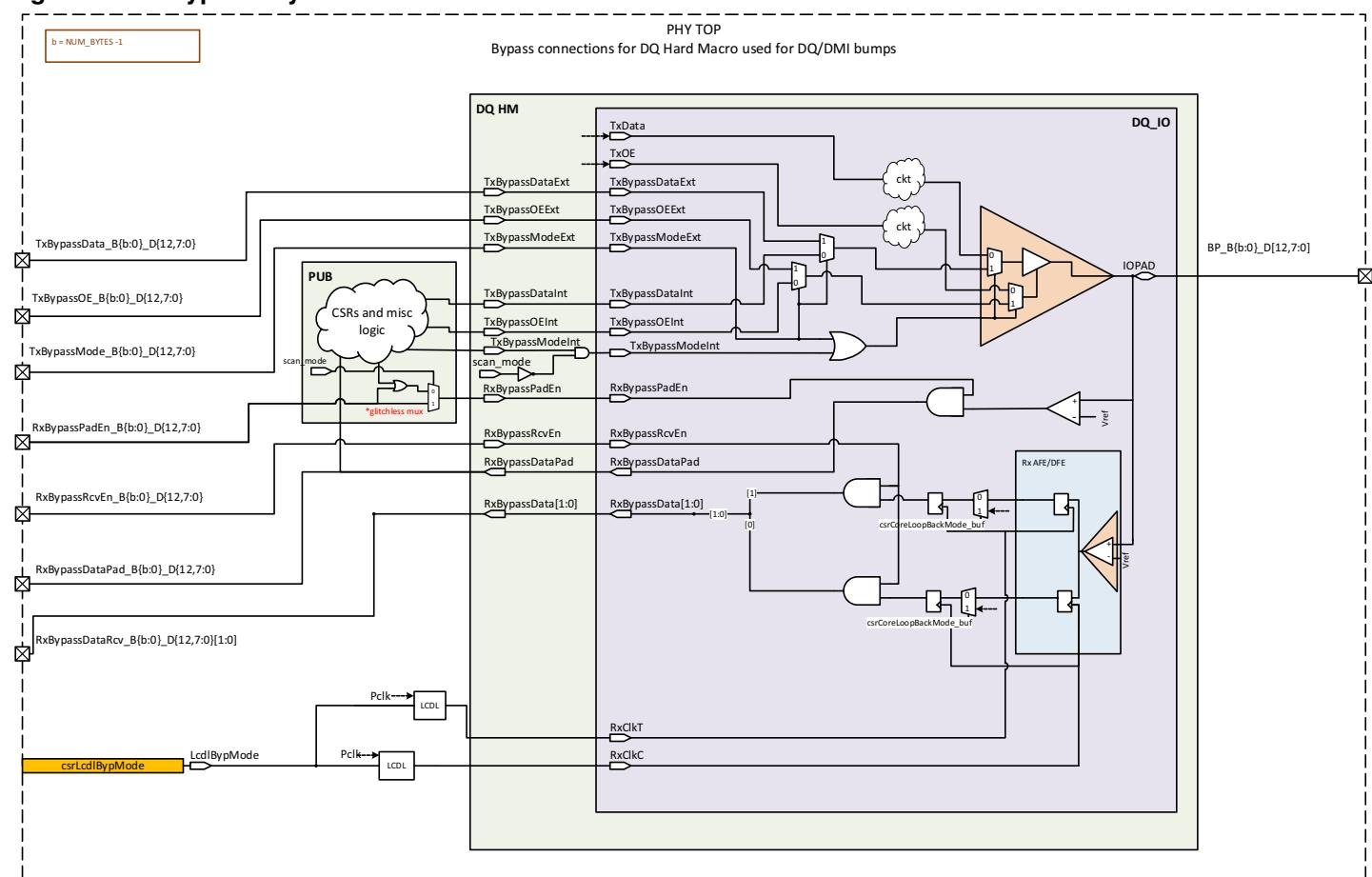
**Table 9-18 BypassMode Control with SCAN\_Mode**

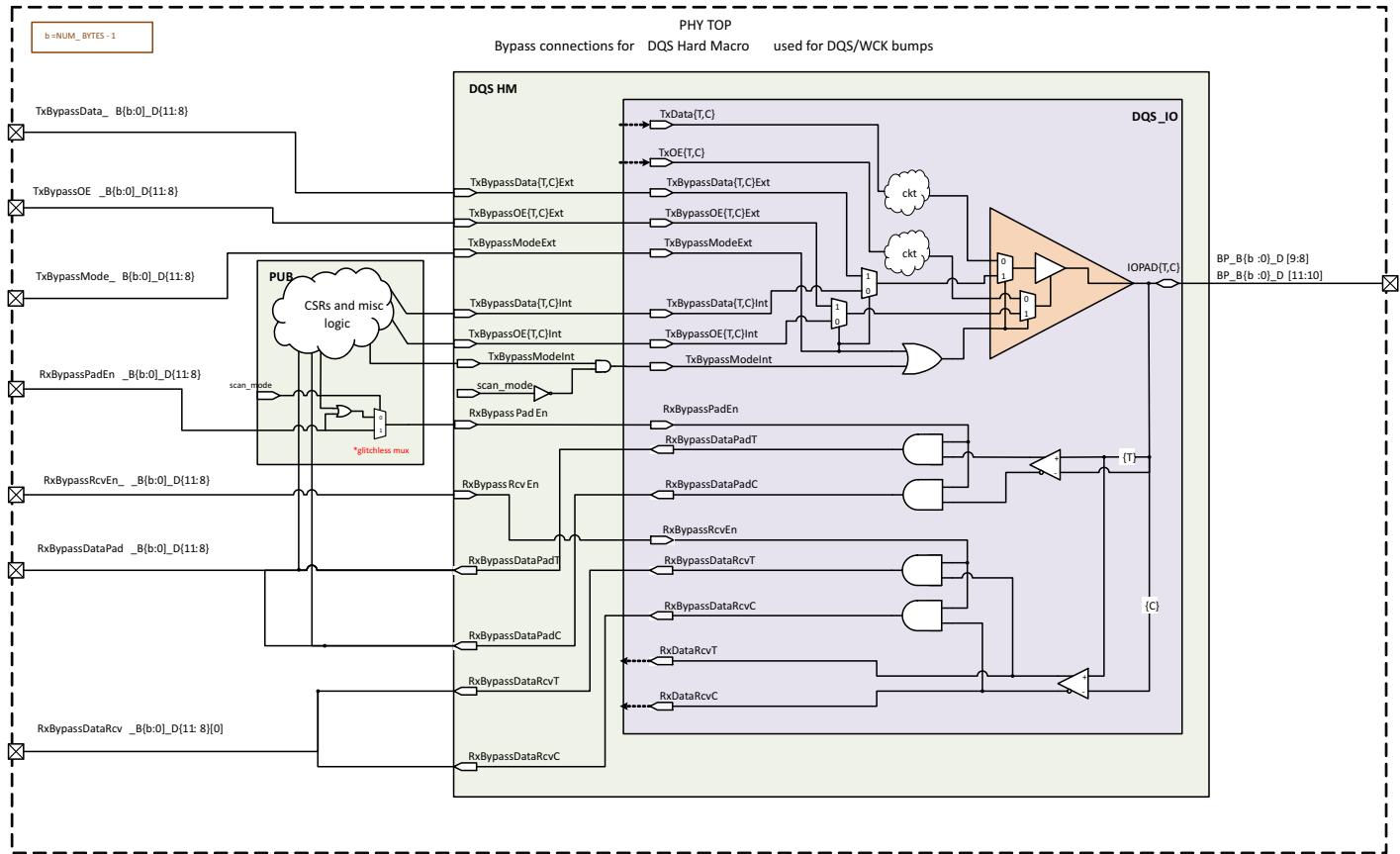
dfiClk	BP_PWROK	SCAN_MODE	IDDQ_MODE	Reset	PHY I/O Pins
X	0	X	X	X	Power-on Defaults (retention mode)
X	1	1	0	X	Controlled by Bypass Mode <sup>a</sup>

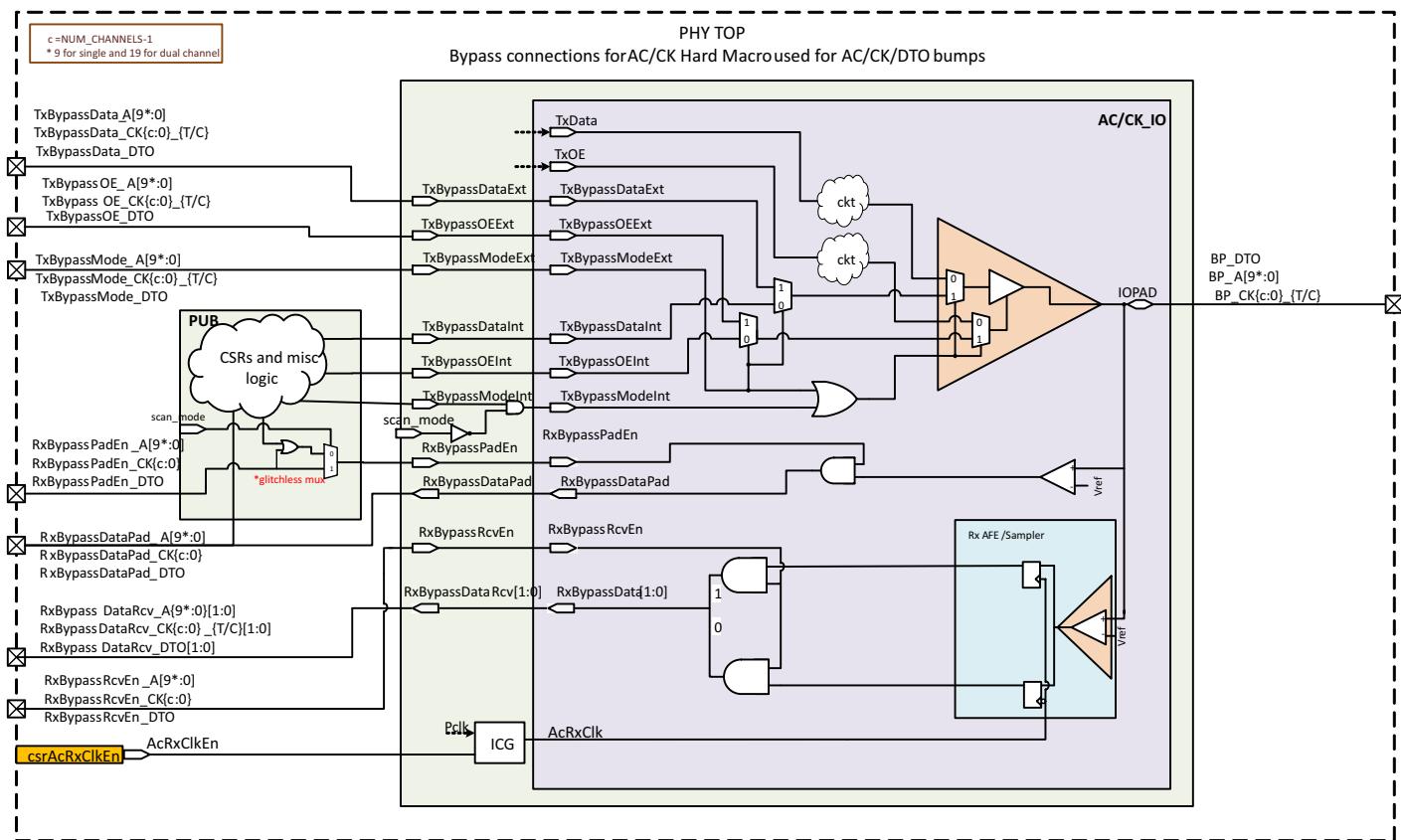
a. BP\_MEMRESET\_L is only controlled in output mode

## 9.7.2 Bypass Mode Architecture

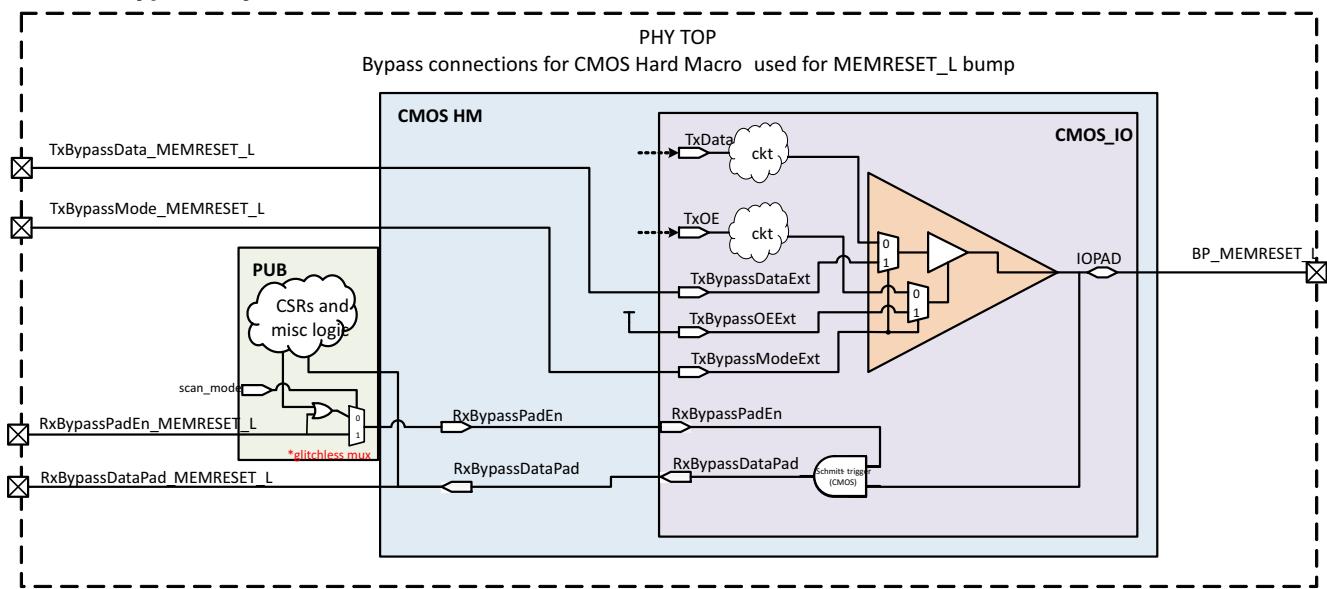
The functional architecture of the Bypass/Flyover controls is shown in the following figure. These ports can be controlled directly from PHY pins or through CSRs.

**Figure 9-5 Bypass/Flyover Controls 1**

**Figure 9-6 Bypass/Flyover Controls 2**

**Figure 9-7 Bypass/Flyover Controls 3**

- See “[PHY AC Configuration](#)” on page 54, BP\_A[9:0] for single Channel, BP\_A[19:0] for dual channel
- BP.DTO exists only when `DWC_LPDDR5XPHY.DTO_ENABLED` is defined.

**Figure 9-8 Bypass/Flyover Controls**

## 9.8 I/O Cell DC Parametric Characterization

The PHY I/O library includes the following features related to IO Cell Characterization:

- Programmable input termination (ODT)
- Programmable output drive impedance
- PVT-compensated ODT and output impedance
- Bypass path to directly control the output driver and monitor received data.
- Receiver power-down control
- PAD and internal loopback modes

The SSTL I/O and PHY Utility Block (PUB) together provide the I/O test functionality. Because control of the testing is via the programmable registers of the PUB, the test methodology may be tailored to suit device requirements. For example, sample testing may be chosen for production testing to reduce test time and more exhaustive testing being reserved for sample/QA testing.

### 9.8.1 Overview

SSTL I/O testing and characterizing includes the following elements:

- General input and output functional operation
- Output drive impedance values and control
- Input termination impedance values and control
- Voltage levels (VOH/VOL/VIH/VIL)

Of these tests, general input and output would be both a production test and a characterization test that would be exercised on all signal pins. The remaining tests are characterization tests that also carry high value in production testing. It is recommended that these tests be run in early production testing on all pins if possible. The amount of testing may be eventually reduced with a better understanding of the foundry process yield and the required operating points for the end product.

In order to test the I/O cells, it is required to have monitor and controllability of following at the system level:

- Bypass mode signals
- External (PAD) pin
- Configuration signals

#### 9.8.1.1 Bypass Mode Signals

See “[Bypass Mode](#)” on page [278](#), for general architecture of bypass mode and a detailed list of signals. Control capability is required for the bypass mode input signals and monitor capability for the output signals. It is recommended to control these signals via standard JTAG boundary scan cells.

### 9.8.1.2 External (PAD) pin

The PAD pin must be controlled and observable via the tester. Additionally, BP\_ZN should be connected to ground with RZN resistor.

### 9.8.1.3 Configuration Registers

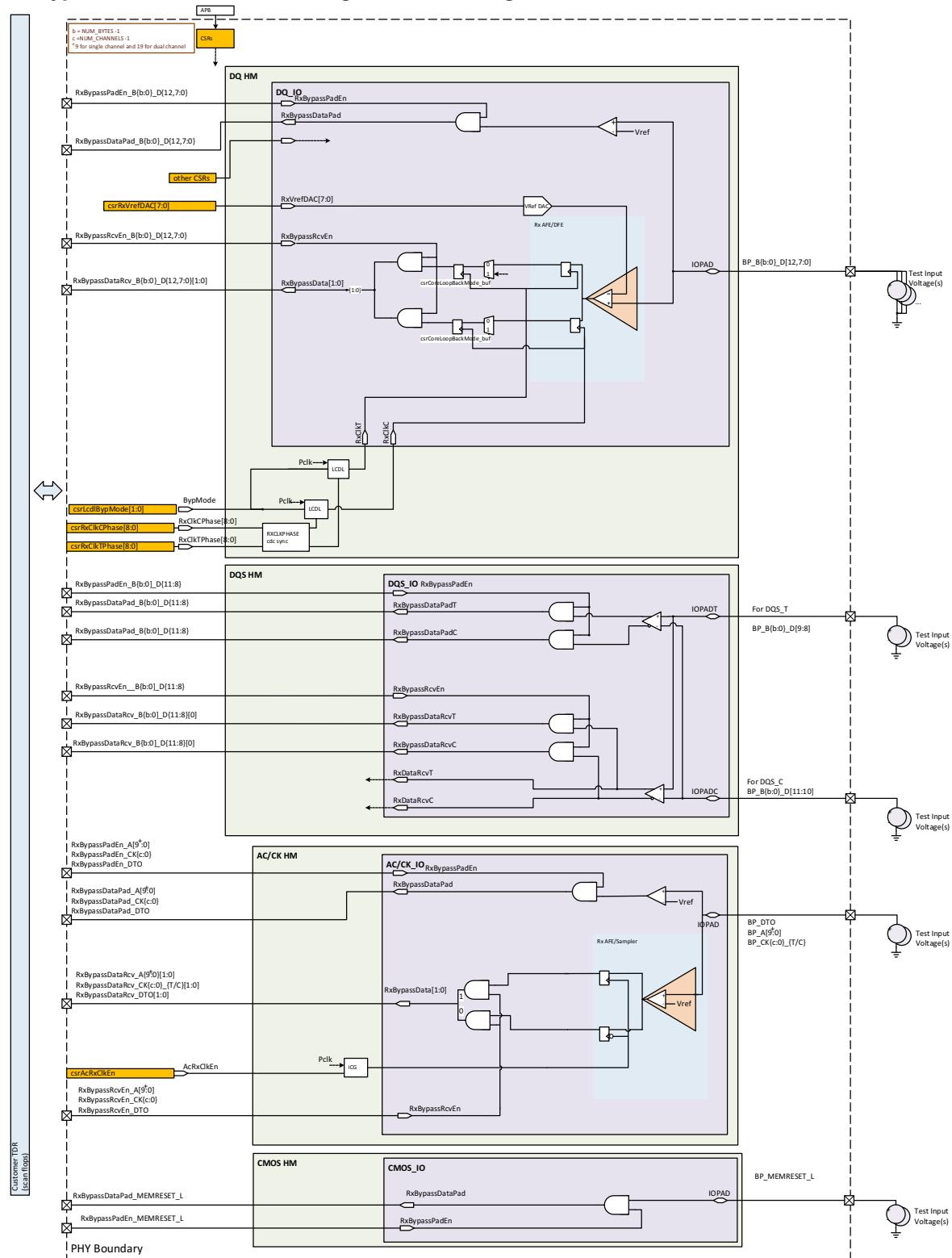
Access to the APB interface is required to read and write configuration registers. The programming must be in sequence according the test instructions in the following sections.

## 9.8.2 VIH/VIL Testing

The VIH/VIL test can be done on all of the DQ/DQS/WCK/DMI/CA/CK.DTO/MEMRESET\_L bit slices independently and in parallel, except in LPDDR5 mode for the restriction of one common RxAcVrefDAC[7:0] being shared between CS and other AC lanes. See the following section for more detail.

Generally, there are two sets of VIH/VIL testing, one for testing out VIH/VIL on bypass/flyover paths, another for testing out VIH/VIL on mission mode receiver paths. CMOS hard macro only supports bypass/flyover paths captured by Schmitt trigger. The diagrams below show the options of two paths.

[Figure 9-9](#) captures the entire bypass path in the PHY for this test.

**Figure 9-9 Bypass Path in the PHY During VIH/VIL Testing**

### 9.8.2.1 Selecting VREF Source

There is an independent local source for each receiver VREF voltage: Local VREF generated through DAC in each DQ/DQS bit slice. For AC bits, a common DAC is used to generate one VREF shared among all AC bit slices.

The output of each receiver is sent out in RxBypassDataRcv\_\* pin of the PHY and it is expected to be captured through boundary scan or shifted in to a TDR outside the PHY for observability.

To perform VIH/VIL testing the tester is required to sweep the voltages of BP\_\* pins for a given static VREF setting. The local VREF is generated through a DAC in each bit slice and the test is required to program the PHY DAC's through the APB interface to generate the desired VREF level.

Each DQ and DQS receiver has its own VREF DAC. VREF DAC in the DQS receiver is intended for singled ended mode of operation if chosen. DqRxVrefDac[8:0] CSRs must be used to program the local DAC for each DQ each receiver, RxDQSSeVrefDAC0[8:0] CSR is used for DQS receiver when in single-ended mode. The output voltage from DAC supports full range from VDDQ to VSS. Nominally the output voltage can be calculated below:

- $V_{ref} (V) = V_{DDQ} (V) * DqRxVrefDac[8:0] / 512$  for DQ
- $V_{ref} (V) = V_{DDQ} (V) * RxDQSSeVrefDAC0[8:0] / 512$  for DQS (single-ended mode)

For AC receivers, a common VREF DAC is used to provide one VREF voltage for all AC bits, RxAcVrefDac[7:0] CSR must be used along with RxAcVrefDacEn to be 1'b1 to program the DAC properly. The output voltage from this DAC supports full range from VDDQ or VDD2H to VSS. Nominally the output voltage can be calculated below:

- $V_{ref} (V) = V_{DD2H} (V) * RxAcVrefDac[7:0] / 256$

Due to the reason of overvoltage protection, the maximum allowable RxAcVrefDac[7:0] should not exceed d'203.

- In LPDDR5X/5 mode, VrefDAC is powered under VDD2H (nominal 1.05 V). While BP\_CS\* lanes are also powered under VDD2H, all other AC lanes are powered under VDDQ. So for VIH/VIL testing, Vref level for CS and other AC lanes need to be adjusted independently. In other words, CS and other AC lanes cannot be tested at the same time due to one common RxAcVrefDac[7:0] is shared among them. However, it may be tested along with other DQ/DQS lanes in parallel.
- In LPDDR4x mode, VrefDAC is powered under VDD2H (nominal 1.10v). While CKE lanes are also powered under VDD2H, all other AC lanes are powered under VDDQ (0.60v), So for VIH/VIL testing, Vref level for CKE and AC/CS lanes need to be adjusted independently. In other words, CKE and AC/CS lanes can't be tested at the same time due to one common RxAcVrefDac[7:0] is shared among them. However, it may be tested along with other DQ/DQS lanes in parallel.

## 9.8.2.2 VIH/VIL Test Sequence

This step is common to DC characterization tests described below and must be followed prior to the sequences described in the next section. Refer to Technology specific PHY databook Electrical Parameters for specific VIH/VIL voltage requirements and PHY databook Register Descriptions for control setting details.

### 9.8.2.2.1 DQ, DMI, CA, CK Test Sequence

To characterize DQ, DMI and CA and CK receivers, do:

1. Bring up VDD, VDDQ, VDD2H and VAA (see “[\(A\) Bring up VDD, VDDQ, VDD2H and VAA](#)” on page [153](#))
2. Start Clocks and Reset the PHY (see “[\(B\) Start Clocks and Reset the PHY](#)” on page [153](#))
3. Put PLL in bypass mode and provide the clock source from PllBypClk pin, no frequency restriction.
  - a. Set csrCPlkBypassMode = 1
  - b. Set csrPclkEn0= 2'b11, csrPclkEn1= 2'b11, and csrClockingCtrl = 2'b00 to enable all Pclk outputs from PAC hard macro.
4. Set PHY pin scan\_mode=0x0
5. Set PHY pin iddq\_mode=0x0
6. Perform the following REGISTER write, if it is not done. Skip this if this has been done from other test sequences.
  - a. Write csrPwrOkDlyCtrl = 0x1 to assert power OK at VDD, VDDQ, VDD2H and VAA domain.
7. Perform the following REGISTER write:
  - a. Program Receiver Power Down and Standby CSR for each bump signal being tested
    - Power Down: {DxRxPowerDownLn\* (for DQ/DMI) or RxPowerDownAC = 1'b0(for AC)}
    - Standby: {DxRxStandbyEn = 1'b0 for DQ/DMI, no standby control for AC}
  - b. For CSX2 testing, program attenuator control appropriately:
    - RxAcAttenCtrlLn\* (for CS hard macro under test) = 0x0
  - c. Program DqRxVrefDac or RxAcVrefDac CSR for each bump signal being tested. For the case of RxAcVrefDaC (for AC), 1ns after it is being programmed, set RxAcVrefDacEn = 1'b1 and wait for 2us.
  - d. Write RxClkT2UIDlyTg\*r\*p\* and RxClkC2UIDlyTg\*\_r\*\_p\* = 0x0 for the bump signals that will be tested, to configure the clock LCDLs (DDL) to the smallest delay value
8. Set PHY in Bypass Mode operation for VIL/VIH testing. (see “[Bypass Mode](#)” on page [278](#) for details of bump mapping and bypass mode operation)
  - a. Set all RxBypassPadEn\* = 0x0
  - b. Set RxBypassRcvEn\* = 0x1 for the bump signals that will be tested;
  - c. Set all TxBypassOE\* = 0x0
  - d. Set all TxBypassMode\* = 0x1
9. Enable the sampling clock through the following CSRs:
  - Assert csrHMBypMode for the DQ and DMI bump signals that will be tested

## 10. Characterize the Receivers for DQ and DMI bumps

- a. Set the voltage level on BP\_\* to the desired level for VIH/VIL. See PHY databook for "VIH/VIL Levels for DQ Receivers" values to the desired VIL/VIH voltage levels.
- b. Wait at least 300 ns.
- c. Sample outputs from the PHY.
  - RxBypassDataRcv\_B{ NUM\_DBYTES-1:0}\_D{12,7:0}[0] - clocked by RxClkC
  - RxBypassDataRcv\_B{ NUM\_DBYTES-1:0}\_D{12,7:0} [1] - clocked by RxClkT
- d. Repeat this step for all the voltage levels to be tested.

## 11. Enable the sampling clock through the following CSRs:

`AcRxClkEn*` = 0x1 for AC and CK signals that will be tested

## 12. Characterize the Receivers for AC and CK bumps.

- a. Set the voltage level on BP\_\* to the desired level for VIH/VIL. See PHY databook for "VIH/VIL Levels for Bypass DQ/DQS Receivers" and "VIH/VIL Levels for Bypass CMOS/CS Receivers" values in both modes, and values for "VIH/VIL Levels for Bypass AC/CK DTO Receivers" in LPDDR4X mode for of desired VIH/VIL voltage levels.
- b. Wait at least 300 ns.
- c. Sample RxBypassDataPad\* outputs from the PHY.
  - RxBypassDataRcv\_A{19:0}[0] -clocked by AcRxClkT
  - RxBypassDataRcv\_A{19:0}[1] - clocked by AcRxClkC
  - RxBypassDataRcv\_CK{1:0}\_{T/C}[ 0 ] -clocked by AcRxClkT
  - RxBypassDataRcv\_CK{1:0}\_{T/C}[1] - clocked by AcRxClkC
- d. Repeat this step for all the voltage levels to be tested.

## 13. Characterize the Flyover receivers for DQ,AC,CK, and DTO bumps

- a. Set all RxBypassRcvEn\* = 0x0
- b. Set RxBypassPadEn\* = 0x1 for the bump signals that will be tested
  - Set the voltage level on BP\_\* to the desired level for VIH/VIL. See PHY databook for "VIH/VIL Levels for Bypass AC/CK DTO Receivers" values.
- c. Wait at least 300 ns.
- d. Sample RxBypassDataPad\* outputs from the PHY.
- e. Repeat this step for all the voltage levels to be tested.

## 14. Complete.



DTO bump is supported only when `DWC_LPDDR5XPHY.DTO_ENABLED` is defined.

### 9.8.2.2.2 DQS, WCK Test Sequence

The BP\_B{NUM\_DBYTES-1:0}\_D[11:10, 9:8]mission receivers directly feed into the clock delay circuits and thus require special programing to enable input differential voltage measurements. Also, WCK pins have a bypass receive path that could be tested. The following sequence should be followed to enable observation of the BP\_B{NUM\_DBYTES-1:0}\_D[11:10, 9:8] receivers depending on the operating mode.

1. Bring up VDD, VDDQ, VDD2H and VAA (see “[\(A\) Bring up VDD, VDDQ, VDD2H and VAA](#)” on page [153](#))
2. Start Clocks and Reset the PHY (see “[\(B\) Start Clocks and Reset the PHY](#)” on page [153](#))
3. Set PHY pin scan\_mode= 0x0
4. Set PHY pin iddq\_mode= 0x0
5. Perform the following REGISTER write if this was not done. Skip this If this has been done from other test sequences.
  - Write csrPwrOkDlyCtrl = 0x1 to assert power OK at VDD, VDDQ, VDD2H and VAA domain.
6. Perform the following REGISTER write:
  - a. Program RxPowerDownDQS = 0 and DxRxStandbyEn = 0 CSR for each DQS/WCK bump signal being tested.
7. Set PHY in Bypass Mode operation for VIH/VIL testing (see “[Bypass Mode](#)” on page [278](#) for details of bump mapping and bypass mode operation)
  - a. Set RxBypassPadEn\*= 0x0 for the bump signals that will be tested, to avoid a high current condition in the CMOS receiver.
  - b. Set RxBypassRcvEn\* = 0x1 for the bump signals that will be tested (Keep RxBypassPadEn\*=0x0).
  - c. Set all TxBypassOE\* = 0x0
  - d. Set all TxBypassMode\* = 0x1
8. Set the BP\_B{NUM\_DBYTES-1:0}\_D[11, 9] (for WCK\_T and DQS\_T) and BP\_B{NUM\_DBYTES-1:0}\_D[10, 8] (for WCK\_C and DQS\_C) bumps to the desired differential voltage to test. See PHY databook for VIH/VIL Levels for DQ Receivers values.
9. Wait at least 300 ns.
10. Sample RxBypassDataRcv\_B{NUM\_DBYTES-1:0}\_D{11,9}[0] (for WCK\_T and DQS\_T) outputs from the PHY. The pin will toggle per the sign of VBP\*\_T - VBP\*\_C given the large enough delta per JEDEC protocol specification.
11. Repeat the previous 4 steps for all the voltage levels to be tested.
12. If so desired, in LPDDR5 mode, place DQS receivers into single-ended mode for testing the case when DQS bumps are used in single-ended fashion.
  - a. RxDiffSeCtrl = 2'b01 for testing DQS\_T/WCK\_T or RxDiffSeCtrl = 2'b10 for testing DQS\_C/WCK\_C bumps.
  - b. Program RxDQSSeVrefDAC0 CSR for each bump signal being tested. 1ns after this is programmed, Set RxDQSDiffSeVrefDACEn = 1'b1.
  - c. Set the voltage level on each DQS\_T/WCK\_T or DQS\_C/WCK\_C bump to the desired level for VIH/VIL. See PHY databook for VIH/VIL Levels for DQ Receivers values reused when testing DQS in single-ended mode.
  - d. Wait at least 300 ns.

- e. Sample RxBypassDataRcv\_B{NUM\_DBYTES-1:0}\_D{12:0}[0] (for WCK\_T and DQS\_T) or RxBypassDataRcv\_B{NUM\_DBYTES-1:0}\_D{12:0}[1] (for WCK\_C and DQS\_C) outputs from the PHY.
  - f. Repeat this step for all the voltage levels and all DQS bumps to be tested.
13. Characterize Flyovers Receivers
- a. Set all RxBypassRcvEn\* = 0x0
  - b. Set RxBypassPadEn\* = 0x1 for the bump signals that will be tested
  - c. Set the BP\_B{NUM\_DBYTES-1:0}\_D[11, 9] (for WCK\_T and DQS\_T) and BP\_B{NUM\_DBYTES-1:0}\_D[10, 8] (for WCK\_C and DQS\_C) bumps to the desired differential voltage to test.  
Refer to the PHY databook for VIH/VIL Levels for Bypass DQ/DQS Receivers.
  - d. Wait at least 300 ns.
  - e. Sample RxBypassDataPad\_B{NUM\_DBYTES-1:0}\_D{11:10,9:8} outputs from the PHY.
  - f. Repeat the 3 previous steps for all the voltage levels to be tested.
14. Complete.

#### 9.8.2.2.3 CMOS Test Sequence

1. Bring up VDD, VDDQ, VDD2H and VAA (see “(A) Bring up VDD, VDDQ, VDD2H and VAA” on page 153)
2. Start Clocks and Reset the PHY (see “(B) Start Clocks and Reset the PHY” on page 153)
3. Set PHY pin scan\_mode=0x0
4. Set PHY pin iddq\_mode=0x1
5. Perform the following REGISTER write if this has not been done already. Skip this if this has been done from other test sequences.
  - Write csrPwrOkDlyCtrl = 0x1 to assert power OK at VDD, VDDQ, VDD2H and VAA domain.
6. Set up bypass receiver for VIH/VIL testing.
  - Set RxBypassPadEn\_MEMORYRESET\_L = 0x1
7. Set the voltage level on BP\_MEMORYRESET\_L to the desired level for VIH/VIL. See PHY databook for values for the desired VIL/VIH voltage levels.
8. Wait at least 300 ns.
9. Sample RxBypassDataPad\_MEMORYRESET\_L output from the PHY.
10. Repeat this step for all the voltage levels to be tested.
11. Complete

### **9.8.3 VOH/VOL Testing**

The VOH/VOL test can be done on all of the DQ/DQS/WCK/DMI/CA/CS/CK/CMOS bit slices independently and in parallel. [Figure 9-10](#) on page 296 captures the entire bypass path in the PHY for this test.

In order to test mission mode drive strength, it is required to connect the BP\_ZN pin to the ground with the same resistance intended to use in mission mode system.

During the test, impedance calibration state machine will run and complete in order to determine the mission mode strength offset so that the desired programmed output impedance is accurately reflected on all PHY drivers.

To make accurate VOL and VOH readings, a Rtt resistor must also be connected to every device pin to V<sub>tt</sub>. Recommended values are: R<sub>tt</sub> = 50 Ohm and V<sub>tt</sub> = VDDQ (or VDD2H)/2. In addition, recommended output driver strength for testing is: 50 Ohm for CS slices, 40 Ohm for all other slices.

During the test, the PHY transmitter drives the BP\_\* to (VDDQ or VDD2H)/VSS with the impedance Rpu/Rdn programmed in the following:

- For Bumps connected to DQ slices:
    - $R_{pu}$  = Impedance specified in TxStrenCodeDqPU\*
  - For Bumps connected to DQS and WCK slices:
    - $R_{pu}$  = Impedance specified in TxStrenCodeDqsPU{T,C}\* $\{T,C\}$
  - For Bumps connected to AC, CK and DTO slices:
    - $R_{pu}$  = Impedance specified in TxStrenCodePUAC\*
  - For Bumps connected to CMOS slices:
    - $R_{pu}$ : refer to PHY databook  $R_{OnCMOSPuBypass}$
    - $R_{pd}$ : refer to PHY databook  $R_{OnCMOSPdBypass}$

$$VOH = [(VDDQ \text{ or } VDD2H) * Rtt + Vtt * Rpu] / (Rpu + Rtt) - X$$

Where X:

- 80 mV for CS slices in LPDDR5X/5 mode, all AC/CK.DTO slices used in LPDDR4x mode, and CMOS slices used in both modes
  - 40 mV for all slices in LPDDR5 mode except CMOS and CS, and all DQ/DQS slices in LPDDR4X mode

DTO bump is supported only when DWC\_LPDDR5XPHY DTO ENABLED is defined.

For CS slices and CMOS slices, the power supply to such slices are VDD2H. VDD2H = 1.05v in LPDDR5 mode and 1.1 v in LPDDR4X nominally.

- For Bumps connected to DQ slices:
    - $R_{pd}$  = Impedance specified in TxStrenCodeDqPD\*
  - For Bumps connected to DQS and WCK slices:
    - $R_{pd}$  = Impedance specified in TxStrenCodeDqsPD{T,C}\*  
T = Tck or Tdata, C = Cck or Cdata
  - For Bumps connected to AC, CK and DTO slices:

- Rpd = Impedance specified in TxStrenCodePDAC\*
- For Bumps connected to CMOS slices:

- Rpd: refer to PHY databook  $R_{OnCMOSpDByPass}$

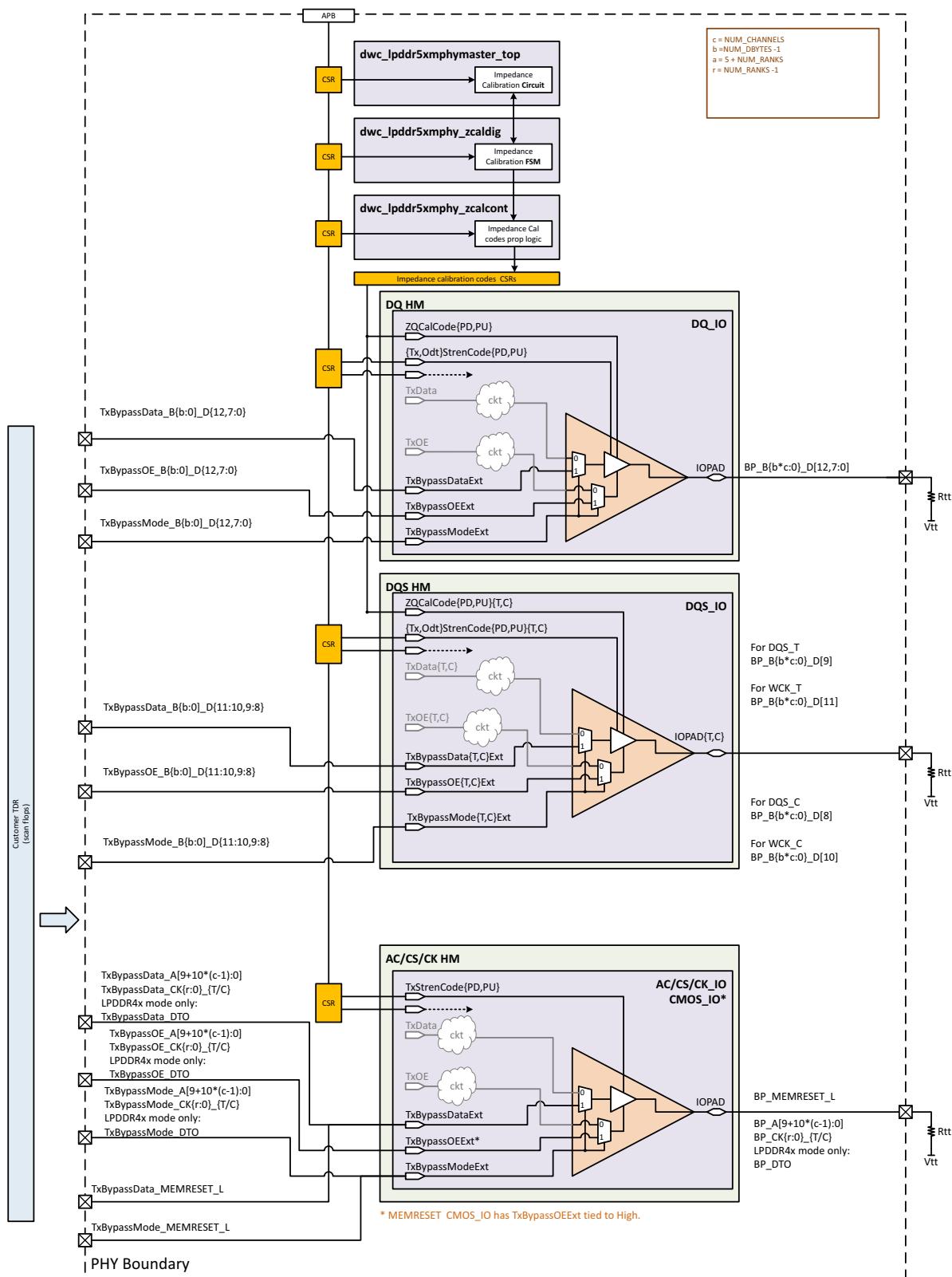
$$VOL = [(Vtt * Rpd) / (Rpd + Rtt)] + Y$$

Where Y:

- 30 mV for CS slices in LPDDR5X/5 mode, and all AC/CK.DTO slices used in LPDDR4x mode
- 15 mV for all slices in LPDDR5 mode except CMOS and AC slices used as CS pins, and all DQ/DQS slices in LPDDR4X mode
- 40 mV for CMOS slices in both modes.

Notice the actual Rpu and Rpd impedance may be varied from the ideal values. For the variation range of Rpu and Rpd, refer to the "Common Output Conditions" in the PHY databook.

[Figure 9-10](#) shows the overall system setup for VOL and VOH testing.

**Figure 9-10 VOH/VOL Testing**

### 9.8.3.1 VOH/VOL Test Sequence

1. Bring up VDD, VDDQ, VDD2H and VAA (see “(A) Bring up VDD, VDDQ, VDD2H and VAA” on page 153)
2. Start Clocks and Reset the PHY (see “(B) Start Clocks and Reset the PHY” on page 153)
3. Set PHY pin scan\_mode = 0x0.
4. Set PHY pin iddq\_mode=0x0
5. Perform the following REGISTER write if this was not done. Skip this If this has been done from other test sequences.
  - a. Write csrPwrOkDlyCtrl = 0x1 to assert power OK at VDD, VDDQ, VDD2H and VAA domain.
6. Perform the following REGISTER write:
  - a. Write TxStrenCodeDq{PU,PD}\* with the desired pull-up and pull-down drive impedances of DQ hard macro.
  - b. Write TxStrenCodeDqs{PU,PD}{T,C}\* with the desired pull-up and pull-down drive impedances of DQS hard macro.
  - c. Write TxStrenCodeAc{PU,PD}\* with the desired pull-up and pull-down drive impedances of AC/CK hard macro.
  - d. Write ZCalCompVRefDAC and ZCalDacRangeSel CSRs to specify the reference voltage to be used by the impedance calibration circuit. See CSR description for more detail
7. Write the following CSRs

**Table 9-19 ODTSeg120 CSR Settings**

DDR Mode	csrOdtSeg120 PU0[3:0]	csrOdtSeg120 PD0[3:0]	csrOdtSeg120 PU1[3:0]	csrOdtSeg120 PD1[3:0]
LPDDR5X/5/4X	0001	0000	0001	0100

**Table 9-20 ZCalTxSlew, and ZCalCompVrefDAC CSR Settings**

DDR Mode	Target VOH	ZCalCompVRefDac	csrTxSlewPU[2:0]	csrTxSlewPD[3:0]
LPDDR5X/5/4X	50%	38	000	0000
LPDDR4X	60%	90	100	0000

- a. Write csrZCalDfiClkTicksPer1uS CSR, to indicate the number of DfiClk ticks per 1us.
- b. Write all csrZCal\*Search\*Auto = 0x0
- c. csrZCalReset = 0x1
- d. csrZCalOnce = 1'b1;
- e. csrZCalReset = 1'b0;
- f. csrZCalClrFirstRunDone = 1'b1
- g. csrZCalClrFirstRunDone = 1'b0

8. Set PHY in Bypass Mode operation for VOL/VOH testing (see "[Bypass Mode](#)" on page [278](#))
  - a. Set TxBypassMode\* = 0x1 for the bump signals that will be tested
  - b. Set all TxBypassOE\* = 0x0
  - c. Set all RxBypassPadEn\* = 0x0
  - d. Set all RxBypassRcvEn\* = 0x0
9. Run impedance calibration. Perform the following REGISTER write via PHY APB interface.
  - a. Kick off calibration by writing ZCalRun = 0x1.
  - b. Wait calibration to end
  - c. Monitor ZCalBusy Or wait tZCAL
  - d. Write ZCalRun = 0x0
  - e. Set ZQUpdate = 0x0
  - f. Set ZQUpdate = 0x1 to propagate the calibrated impedance codes to the transmitters
  - g. Set ZQUpdate = 0x0
10. Wait 16 DfiClks, and an additional 400ns for codes to stabilize
11. Set up the tester to connect a resistor Rtt between the BP\_\* bumps and Vtt.
12. Drive the following PHY pins:
  - a. To measure VOH: TxBypassData\* = 0x1, for the bump signals that will be tested
  - b. To measure VOL: TxBypassData\* = 0x0, for the bump signals that will be tested
  - c. Wait 1 ns
  - d. TxBypassOE\* = 0x1 for the bump signals that will be tested
13. Measure voltage at BP\_\* and compare with expected result from formula at the tester to determine pass/fail criteria.
14. Run impedance calibration before any measurement, every time VT changes, repeating step 8.
  - This is not needed for characterization done to the CMOS IO as this is not calibrated.
  - This is not needed for characterization done to the CS pins in LPDDR5X/5 mode as these are not calibrated.

#### 9.8.4 ODT Impedance Testing

The ODT Impedance test can be done on all of the DQ and DQS bit slices independently and in parallel. [Figure 9-10](#) on page [296](#) captures the entire bypass path in the PHY for this test.

In order to test mission mode drive strength, it is required to connect the BP\_ZN pin to the ground with the same resistance intended to use in mission mode system.

During the test, impedance calibration state machine will run and complete in order to determine the mission mode strength offset so that the desired programmed output impedance is accurately reflected on all PHY drivers.

To make accurate readings, VI\_PAD should be connected to VDDQ/2.

During the test, the PHY transmitter drives the BP\_DQ\* to (VDDQ)/VSS with the impedance programmed in the following:

- For Bumps connected to DQ slices:
  - Impedance specified in csrOdtStrenCodeDqPD\*
    - csrOdtStrenCodeDqPU\* should be set to 0
- For Bumps connected to DQS slices:
  - Impedance specified in OdtStrenCodeDqsPD{T,C}\*
    - csrOdtStrenCodeDqsPU\* should be set to 0

Notice the actual impedance may be varied from the ideal value. For the variation range, refer to the Common Output Conditions in the PHY databook for more detail.

#### 9.8.4.1 ODT Impedance Test Sequence

Following are instructions to setup the PHY to measure ODT impedance.

- Ensure that a 120 Ohms, 1% max tolerance, external reference resistor is connected at BP\_ZN bump pin (if other resistor values are in use, contact Synopsys for further instructions).
1. Bring up VDD, VDDQ, VDD2H and VAA (see “(A) Bring up VDD, VDDQ, VDD2H and VAA” on page 153)
  2. Start Clocks and Reset the PHY (see “(B) Start Clocks and Reset the PHY” on page 153)
  3. Set PHY pin scan\_mode = 0x0
  4. Set PHY pin iddq\_mode=0x0
  5. Perform the following REGISTER write if this is not done. Skip this If this has been done from other test sequences.
    - a. Write csrPwrOkDlyCtrl = 0x1 to assert power OK at VDD, VDDQ VDD2H and VAA domain.
  6. Perform the following REGISTER write:
    - a. Write ZCalCompVRefDAC and ZCalDacRangeSel CSRs to specify the reference voltage to be used by the impedance calibration circuit. See CSR description for more detail.
    - b. Write the following CSRs.

**Table 9-21 ODTSeg120 CSR Settings**

DDR Mode	csrOdtSeg120PU0 [3:0]	csrOdtSeg120PD0 [3:0]	csrOdtSeg120PU1 [3:0]	csrOdtSeg120PD1 [3:0]
LPDDR5X/5/4X	0001	0000	0001	0100

**Table 9-22 ZCalTxSlew, and ZCalCompVrefDAC CSR Settings**

<b>DDR Mode</b>	<b>Target VOH</b>	<b>ZCalCompVRefDac</b>	<b>csrTxSlewPU[2:0]</b>	<b>csrTxSlewPD[3:0]</b>
LPDDR5X/5/4X	50%	38	000	0000
LPDDR4X	60%	90	100	0000

- c. Write csrZCalDfiClkTicksPer1uS CSR, to indicate the number of DfiClk ticks per 1us.
- d. Write all csrZCal\*Search\*Auto = 0x0
- e. csrZCalReset = 0x1;
- f. csrZCalOnce = 1'b1;
- g. csrZCalReset = 1'b0;
- h. csrZCalClrFirstRunDone = 1'b1
- i. csrZCalClrFirstRunDone = 1'b0
- j. Write csrZCalCompVRefDAC, to specify the reference voltage to be used by the impedance calibration circuit. Refer to PHY databook
- k. Write csrDxDtEn[10:0]=10'h7ff
- 1. Run impedance calibration. Perform the following REGISTER write via PHY APB interface.
- 7. Kick off calibration by writing ZCalRun = 0x1
  - a. Wait calibration to end
  - b. Monitor ZCalBusy Or wait tZCAL
  - c. Write ZcalRun = 0x0
  - d. Write ZQUpdate = 0x0
  - e. Write ZQUpdate = 0x1 to propagate the calibrated impedance codes to the transmitters
  - f. Write ZQUpdate = 0x0
  - g. Wait 16 DfiClks, and an additional 400ns for codes to stabilize
- 8. Set up the tester to connect a resistor Rtt between the BP\_\* bumps and Vtt.
- 9. Program OdtImpedance\* CSRs with the desired impedance values for the corresponding IOs.
  - Note: PHY ODT applies to DQ and DQS pins only
    - a. Wait 16 DfiClks
    - b. Wait 2000 ps
- 10. Measure the termination impedance at the BP\_\* pins, for the target DRAM's voltage.

## 9.9 Quiescent Current (IDQ) Measurement

To simplify the test effort, a single-pin, IDQ\_mode, from the PHY\_TOP is utilized. However, in order to provide an accurate measurement, the following pins must be driven:

- Set PHY\_TOP input BP\_PWROK=1
- Ensure all clocks are stopped for current measurement
  - Set PHY\_TOP input DfiClk to 0
  - Set PHY\_TOP input APBCLK to 0
  - Set PHY\_TOP inputs scan\_clk to 0
  - Set PHY\_TOP Bypass Mode Inputs to 0
- Set iddq\_mode=1
  - When IDQ\_mode = 1, the following actions are taken to ensure circuits in both VDD and VDDQ/VDD2H domain are in quiescent state for IDQ current measurement.
- PHY will turn off DAC Vref Generator.
- PHY will put PLL in power down state
- VDDQ IOs are in lowest power state
  - PHY will tristate DQ/DQS pins
  - PHY will tristate AC/CLK/ZN/CS(LPDDR5)/CKE(LPDDR4X)
  - PHY will tristate BP\_ATO, BP\_ATO\_PLL, and BP.DTO
  - If scan\_mode = 0, PHY will tristate BP\_MEMRESET
  - If scan\_mode = 1, PHY will set BP\_MEMRESET=0
- PHY will disable Impedance calibration.
- PHY will stop LCDL Ring oscillator clocks.

In this state, quiescent current for each power rail can be measured 300 ns after PHY pin iddq\_mode=1. IDQ\_mode in combination of scan\_mode configuration offers different states of operations. This is outlined in the table below:

**Table 9-23 IDQ\_mode vs Scan\_mode configuration with states of operation**

IDQ_mode	Scan_mode	State of Operation
0	0	Mission mode operation
0	1	Scan mode operation
1	0	Quiescent current (IDQ) measurement
1	1	Quiescent current (IDQ) measurement with scan enabled

## 9.10 Input Pin Leakage Measurement

To perform Input Pin Leakage measurements, the following PHY top-level input signals must be set:

- TxBypassMode\_A[18:0] = All Ones
- TxBypassOE\_A[18:0] = All Zeros
- TxBypassMode\_B{ NUM\_DBYTES-1:0}\_D{12:0} = All Ones
- TxBypassOE\_B{ NUM\_DBYTES-1:0}\_D{12:0} = All Zeros
- TxBypassMode\_CK{3:0}\_{T,C} = All Ones
- TxBypassOE\_CK{3:0}\_{T,C} = All Zeros
- TxBypassMode.DTO = All Ones
- TxBypassOE.DTO = All Zeros
- Iddq\_mode = 1
- IO pad voltage level should be set to VDD voltage level or lower

All PHY I/O pins will be tristated in this mode and will support the input pin leakage measurements.

The PHY output-only pin BP\_MEMRESET\_L does not support input pin leakage measurements (or the JTAG HIGHZ command).

The PHY Analog pins { BP\_ZN, BP\_ATO, BP\_ATO\_PLL, BP\_PWROK } do not support input pin leakage measurements.



This state is typically achieved when the customer has connected the PHY Bypass\* signals to a TDR that implements the JTAG HIGHZ command.

# 10

## Timing

The following sections are included in this chapter:

- “PHY Latency” on page 304
- “PclkPtrInitVal Impacts From Construction Skews” on page 305
- “Rank-to-Rank Spacing” on page 306
- “DQ/DQS Bus Turnarounds” on page 307
- “Preamble and Postamble Restrictions” on page 311
- “PHY Skew Limits” on page 312
- “DFI Low Power Interface” on page 316
- “DFI Update Interface” on page 323
- “DFI Error Interface” on page 326
- “DFI Master Interface” on page 328
- “PHY DFI Sideband Interactions” on page 329
- “DFI Sideband Collisions” on page 333

## 10.1 PHY Latency

The maximum read latency on the DFI interface (DFIMRL) and the  $t_{phy\_rdlat}$  parameter has a defined relationship that is outlined as below:

**Table 10-1 Read Latency LPDDR5 1:2 Mode**

From	To	$t_{phy\_rdlat}$ (in Wck)
dfi_rddata_en_P0	dfi_rddata_valid_W0	$(6 + csrDFIMRL)^*2 + (DWC\_LPDDR5XPHY\_PIPE\_DFI\_RD^*2 + csrDxInPipeEn + csrDxOutPipeEn)^*2$
dfi_rddata_en_P1	dfi_rddata_valid_W0	$(6 + csrDFIMRL)^*2 + (DWC\_LPDDR5XPHY\_PIPE\_DFI\_RD^*2 + csrDxInPipeEn + csrDxOutPipeEn)^*2 - 1$

**Table 10-2 Read Latency LPDDR5 1:4 Mode**

From	To	$t_{phy\_rdlat}$ (in Wck)
dfi_rddata_en_P0	dfi_rddata_valid_W0	$(6 + csrDFIMRL)^*4 + (DWC\_LPDDR5XPHY\_PIPE\_DFI\_RD^*2 + csrDxInPipeEn + csrDxOutPipeEn)^*4$
dfi_rddata_en_P1	dfi_rddata_valid_W0	$(6 + csrDFIMRL)^*4 + (DWC\_LPDDR5XPHY\_PIPE\_DFI\_RD^*2 + csrDxInPipeEn + csrDxOutPipeEn)^*4 - 1$
dfi_rddata_en_P2	dfi_rddata_valid_W0	$(6 + csrDFIMRL)^*4 + (DWC\_LPDDR5XPHY\_PIPE\_DFI\_RD^*2 + csrDxInPipeEn + csrDxOutPipeEn)^*4 - 2$
dfi_rddata_en_P3	dfi_rddata_valid_W0	$(6 + csrDFIMRL)^*4 + (DWC\_LPDDR5XPHY\_PIPE\_DFI\_RD^*2 + csrDxInPipeEn + csrDxOutPipeEn)^*4 - 3$

**Table 10-3 Read Latency LPDDR4X 1:2 Mode**

From	To	$t_{phy\_rdlat}$ (in MEMCLK)
dfi_rddata_en_P0	dfi_rddata_valid_W0	$(6 + csrDFIMRL)^*2 + (DWC\_LPDDR5XPHY\_PIPE\_DFI\_RD^*2 + csrDxInPipeEn + csrDxOutPipeEn)^*2$
dfi_rddata_en_P1	dfi_rddata_valid_W0	$(6 + csrDFIMRL)^*2 + (DWC\_LPDDR5XPHY\_PIPE\_DFI\_RD^*2 + csrDxInPipeEn + csrDxOutPipeEn)^*2 - 1$

**Table 10-4 Read Latency LPDDR4X 1:4 Mode**

From	To	$t_{phy\_rdlat}$ (in MEMCLK)
dfi_rddata_en_P0	dfi_rddata_valid_W0	$(6 + csrDFIMRL)^*4 + (DWC\_LPDDR5XPHY\_PIPE\_DFI\_RD^*2 + csrDxInPipeEn + csrDxOutPipeEn)^*4$
dfi_rddata_en_P1	dfi_rddata_valid_W0	$(6 + csrDFIMRL)^*4 + (DWC\_LPDDR5XPHY\_PIPE\_DFI\_RD^*2 + csrDxInPipeEn + csrDxOutPipeEn)^*4 - 1$
dfi_rddata_en_P2	dfi_rddata_valid_W0	$(6 + csrDFIMRL)^*4 + (DWC\_LPDDR5XPHY\_PIPE\_DFI\_RD^*2 + csrDxInPipeEn + csrDxOutPipeEn)^*4 - 2$
dfi_rddata_en_P3	dfi_rddata_valid_W0	$(6 + csrDFIMRL)^*4 + (DWC\_LPDDR5XPHY\_PIPE\_DFI\_RD^*2 + csrDxInPipeEn + csrDxOutPipeEn)^*4 - 3$

Where:

- DFIMRL: DFI Max Read Latency (units of DFI Clocks) Programmed in DBYTE Csr Slave
- $t_{phy\_rdlat}$  Phase n: when dfi\_rddata\_en is launched on Phase n



**Note** The PHY returns dfi\_rddata\_valid contiguously for each burst. There are no bubbles inserted within a burst.

### 10.1.1 Max Read Latency (DFIMRL)

DFIMRL is a trained parameter of the PHY. The training firmware includes a user-supplied margin value which is added to the smallest passing trained DFIMRL to form the final DFIMRL value. The margin value includes the maximum expected drift of tDQSCK (due to voltage and temperature), and factors in the worst-case user application. DFIMRL compensates for many elements including PHY IO cell analog delay, MEMCLK flight time from PHY to furthest rank, DQS flight time from furthest rank to PHY and tDQSCK from DRAM.

When in DFI 1:4 Mode, the user-supplied margin value, DFIMRL\_Margin, must be set to 1

### 10.1.2 PclkPtrInitVal Impacts From Construction Skews

The value of PclkPtrInitVal is constrained by the implementation of the PHY hard macro of the PHY hard macros, DFI {PUB, TUB HM} clock period, clocking mode (1:2/1:4) and DDR type (LPDDR4X/LPDDR5/LPDDR5X) The value of the PclkPtrInitVal must be set based on the following table.

**Table 10-5 PclkPtrInitVal Pointer Separation Ranges**

Protocol	All	All	LPDDR4X Only	All	LPDDR5/5x	LPDDR5X
Data rate (Mbps)	3200.00	3200.00	4267.00	4267.00	6400.00	9600.00
Ratio (1:2/1:4)	1:2	1:4	1:2	1:4	1:4	1:4
PclkPtrInitVal minimum value (UI)	3	3	3	3	3	3
PclkPtrInitVal maximum value (UI)	10	10	9	10	9	7

PclkPtrInitVal must be increased based on the following variations:

- DfiClk Jitter: Clock jitter from DfiClk generation source to the HardIP DfiClk input ports.
- DfiClk PVT variations: Voltage, temperature and process variations on the clock tree branches from the divergence point to the separate HardIP DfiClk input ports.
- PClk Jitter: Clock jitter from the PLL generating PClk to the HardIP PClk input ports. This should be minimized if the PClk routing guidance in the implementation guide is followed. In this case, jitter refers to the Time Interval Error (TIE) of the DfiClk and Pclk, measured over N cycles, where N is large enough to include all frequencies greater than the PHY PLL bandwidth:
  - $N_{DfiClk} = f_{DfiClk}/PLL\_BW$
  - $N_{Pclk} = f_{Pclk}/PLL\_BW$

Each incremental addition of each of these variations sum together to produce the amount by which the PclkPtrInitVal must be increased to properly space the FIFO pointers. With the unit of PclkPtrInitVal being

in 1 Unit Interval (UI), the chosen value should be the ceiling of the sum of variations divide by the Unit Interval for a particular data rate, refer to the table above.

## 10.2 Rank-to-Rank Spacing

Minimum possible Rank-to-Rank for Write-Write (W-W) transaction:

- LPDDR5 Mode: tphy\_wckcsgap
- LPDDR4X Mode: tphy\_wrcsgap + ceiling[tDQS2DQ\_rank2rank/tCK], where tCK is memory clock period

Minimum possible Rank-to-Rank for Read-Read (R-R) transaction:

- LPDDR5 Mode: tphy\_wckcsgap + ceiling[tWCK2DQO\_rank2rank/tWCK], where tWCK is WCK clock period.
- LPDDR4X Mode: tphy\_rdcsgap + ceiling[tDQSCK\_rank2rank/tCK], where tCK is memory clock period

Minimum possible Rank-to-Rank for Write-Read (W-R) transaction:

- LPDDR5 Mode: tphy\_wckcsgap.
- LPDDR4X Mode: Refer to “[WDQS Extension](#)” on page [225](#).

Minimum possible Rank-to-Rank for Read-Write (R-W) transaction:

- LPDDR5 Mode: tphy\_wckcsgap + ceiling[tWCK2DQO\_rank2rank/tWCK], where tWCK is WCK clock period.
- LPDDR4X Mode: Refer to “[WDQS Extension](#)” on page [225](#).

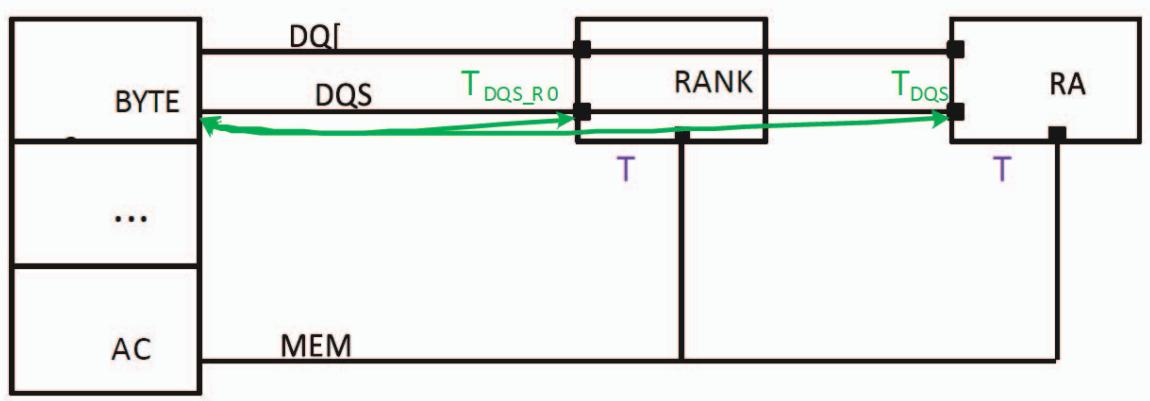
These values reflect the time required by the PHY to perform internal timing setting updates. The achievable rank-rank spacings in a system will depend on the channel topology and the termination enable timing of the target DRAMs, as well as CDD, which is described in “[DQ/DQS Bus Turnarounds](#)” on page [307](#).

For more information on DQS/WCK postamble and preamble options, see “[Preamble and Postamble Restrictions](#)” on page [311](#).

## 10.3 DQ/DQS Bus Turnarounds

The following constraints apply for an unterminated system configuration. Thus, these equations only consider scheduling constraints for the defined toggles/preambles/postambles of DQ/DQS. Additional DRAM ODT scheduling constraints will need to be considered in terminated systems to ensure correct operation.

**Figure 10-1 Example System, and PHY Trained Parameters**



### 10.3.1 DRAM Transaction Timing

The Target Rank[i] for a read transaction will begin transmitting the Read DQS preamble at

- $T_{CK\_R[i]} + RL - N_{PRE} // N_{PRE} = \{1,2\}$

The Target Rank[i] for a write transaction will begin receiving the Write DQS preamble at

- $T_{CK\_R[i]} + WL - N_{PRE} // N_{PRE} = \{1,2\}$

### 10.3.2 PHY Trained State

PHY register RxEnDlyD[i] encodes the timing of the PHY readgate, which asserts in the middle of the preamble:

- $T_{CK\_R[i]} + T_{DQS\_R[i]} + RL - N_{PRE}/2 // N_{PRE} = \{1,2\}$

PHY register TxDQSDlyD[j] encodes the launch timing of the DQS from the PHY, which must be at

- $T_{CK\_R[j]} - T_{DQS\_R[j]} + WL - N_{PRE} // N_{PRE} = \{1,2\}$

### 10.3.3 LPDDR4X Mode CDD

#### [R2W] Read from Rank[i]; Write to Rank[j]

Constraint :: Read DQS postamble received by the PHY should complete before the PHY attempts to transmit the Write DQS preamble.

Since the constraint is measured at the PHY DQS, we can use the differences of the trained PHY CSR values for RxEnD[j] and TxDqsDlyD[j].

The absolute value of the PHY FW returned values CDD\_RW\_R[i]\_R[j] can be used to adjust scheduling of reads to writes.

- $CDD\_RW\_R[i]_R[j] (\text{in UI}) = \text{abs}(RxEnD[i] - TxDqsDlyD[j])$

- $t_{CCDmin}(R_{rank}[i], W_{rank}[j])$  (in MemClks) = System\_R2W + ceiling(CDD\_RW\_R[i]\_R[j]/2)  
System\_R2W = scheduling of read/write transactions after correcting for WL, RL, BL, preamble lengths  
For System\_R2W value, refer to the equation defined in section "[WDQS Extension](#)" on page [225](#).

### [W2R] Write to Rank[i]; Read from Rank[j]

Constraint :: Write DQS transmitted by PHY should be observed to be complete at all Ranks before any Rank attempts to transmit the Read DQS.

- Rank[i] (target rank) will complete receiving the Write DQS postamble at:
  - $T_{CK\_R[i]} + WL + BL$
- Other, non-target ranks will also complete receiving the Write DQS postamble at:
  - $T_{CK\_R[i]} + WL + BL + \max(CDD\_WW\_R[i]_R[0..1])$
- This means the read transaction at any rank can legally start at:
  - $T_{write} + WL + BL + \max(\text{abs}(CDD\_WW\_R[i]_R[0..1]))$
- $CDD\_WR\_R[i]_R[0..1]$  (in UIs) =  $\max[0, (\max \text{ across all bytes}(TxDqsDlyD[i])) - (\min \text{ across all bytes}(TxDqsDlyD[j]))]$
- $t_{CCDmin}(W_{rank}[i], R_{rank}[j])$  (in MemClks) = System\_W2R + ceiling(CDD\_WW\_R[i]\_R[0..1]/2)  
System\_W2R = scheduling of write/read transactions after correcting for WL, RL, BL, preamble lengths.

For System\_W2R value, refer to the equation defined in section "[WDQS Extension](#)" on page [225](#).

At LP4X-4267 Mbps if the trained write leveling delay is larger than  $TxDqsDly > 8UI$ , Write-to-Read command gap needs to be increased by 1tCK.

### [R2R] Read from Rank[i]; Read from Rank[j]

Constraint:: Read DQS postamble received by the PHY should complete for current Rank before next Rank attempts to transmit the Read DQS.

Since the constraint is measured at the PHY DQ, we can use the differences of the trained PHY CSR values for RxEnd[i] and RxEnd[j].

- $CDD\_RR\_R[i]_R[0..1]$  (in UIs) =  $\max[0, (\text{RxEndDly coarse}[i] - \text{RxEndDly coarse}[j])]$
- $t_{CCDmin}(R_{rank}[i], R_{rank}[j])$  (in MemClks) = System\_R2R + ceiling(CDD\_RR\_R[i]\_R[0..1]/2)  
System\_R2R = scheduling of read transactions to different rank after correcting for RL, BL, preamble lengths  
System\_R2R = Baseline + Rank-to-Rank spacing,  
Where Baseline is defined in JEDEC spec. For Rank-to-Rank spacing value, refer to section "[Rank-to-Rank Spacing](#)" on page [306](#). Rank-to-Rank spacing is not required in case of same rank access.

### [W2W] Write to Rank[i]; Write to Rank[j]

Constraint:: Write DQS transmitted by PHY should be observed to be complete at Rank[i] before PHY attempts to transmit Write DQS to Rank[j].

Since the constraint is measured at the PHY Write DQS, we can use the differences of the trained PHY CSR values for TxDqsDlyD[i] and TxDqsDlyD[j].

- $CDD\_WW\_R[i]_R[0..1]$  (in UIs) =  $\max[0, (\max \text{ across all bytes}(TxDqsDlyD[i])) - (\min \text{ across all bytes}(TxDqsDlyD[j]))]$

- $t_{CCDmin}(W_{rank}[i], W_{rank}[j])$  (in MemClks) =  $System\_W2W + \text{ceiling}(CDD\_WW\_R[i]_R[0..1]/2)$   
 $System\_R2R$  = scheduling of read transactions to different rank after correcting for RL, BL, preamble lengths.

$System\_W2W$  = Baseline + Rank-to-Rank spacing,  
Where Baseline is defined in JEDEC spec. For Rank-to-Rank spacing value, refer to section “[Rank-to-Rank Spacing](#)” on page [306](#). Rank-to-Rank spacing is not required in case of same rank access.

#### 10.3.4 LPDDR5 Mode CDD

Constraint :: In LPDDR5 mode, WCK is used by DRAM to capture commands during write as well read mode and must be valid before Wrte DQ or Read DQ can start. Hence CDD is determined by WCK delays.

Command gap [nCK]

= Baseline + Rank-to-Rank spacing + ceiling(CDD/4) (CK:WCK = 1:2 mode)

= Baseline + Rank-to-Rank spacing + ceiling(CDD/8) (CK:WCK = 1:4 mode),

Where Baseline is defined in JEDEC spec and CDD has unit of UI. For Rank-to-Rank spacing value, refer to section “[Rank-to-Rank Spacing](#)” on page [306](#). Rank-to-Rank spacing is not required in case of same rank access.

**Table 10-6 LPDDR5 Mode CDD with WCK Enabled and Disabled**

CDD	WCK always ON Disabled (MR18.OP[4] = 0)	WCK always ON Enabled(MR18.OP[4] = 1)
For Different Rank		
Read rank0 to CAS* rank1	$\max\{(\text{Max across all bytes } (TxWckDlyTg0 - TxWckDlyTg1)), 0\}$	N/A
Write rank0 to CAS* rank1	$\max\{(\text{Max across all bytes } (TxWckDlyTg0 - TxWckDlyTg1)), 0\}$	N/A
Read rank0 to Write rank1	N/A	RDQSt/c Disabled (MR20.OP[1:0] = ‘b00): $\max\{\text{Max across all bytes } (\max \text{ across all lanes } (RxDigStrbDlyTg0\_In<r> - TxDqDlyTg1\_In<r>)), 0\}$ RDQSt/c Enabled: $\max\{\text{Max across all bytes } (RxEnDlyTg0 - \min \text{ across all lane } (TxDqDlyTg1, TxDqsDlyTg1(\text{if WECC}))), 0\}$
Read rank0 to Read rank1	N/A	RDQSt/c Disabled (MR20.OP[1:0] = ‘b00): $\max[0, \text{Max across all bytes } (\max \text{ across all lanes } (RxDigStrbDly coarse\_Tg0 - RxDigStrbDly coarse\_Tg1))]$ RDQSt/c Enabled: $\max[0, \text{Max across all bytes } (RxEnDly coarse\_Tg0 - RxEnDly coarse\_Tg1)]$

CDD	WCK always ON Disabled (MR18.OP[4] = 0)	WCK always ON Enabled(MR18.OP[4] = 1)
Write rank0 to Write rank1	N/A	max {(Max across all bytes (TxDqDlyTg0, TxDqsDlyTg0(if ECC) – TxDqDlyTg1, TxDqsDlyTg1(if ECC))), 0}
Write rank0 to Read rank1	N/A	max {(Max across all bytes (TxDqDlyTg0, TxDqsDlyTg0(if ECC) – TxDqDlyTg1, TxDqsDlyTg1(if ECC))), 0}
For the same rank		
Read rank0 to Write rank0	RDQSt/c Disabled (MR20.OP[1:0] = 'b00): max{Max across all bytes (max across all lanes (RxDigStrbDlyTg0_In<r> – TxDqDlyTg0_In<r>)), 0} RDQSt/c Enabled: max{Max across all bytes(RxEnDlyTg0 – Min across all lane(TxDqDlyTg0, TxDqsDlyTg0(if WECC)), 0}	
Write rank0 to Read rank0	0	

## 10.4 Preamble and Postamble Restrictions

### LPDDR5 Preamble/Postamble rules

The Tphy\_rdcsgap and Tphy\_wrcsgap DFI timing parameters should be 0 in LPDDR5 mode. The Tph\_wckcsgap parameter provides command gap whenever rank switches.

- Read DQS Preamble:
  - Static:2\*tWCK,Toggle:2\*tWCK read preambles are required (MR10.OP[5:4]==2'b01).
  - The default T<sub>phy\_wckcsgap</sub> value covers this setting.
- Read DQS Postamble:
  - The 2.5\*tWCK and 4.5\*tWCK read postambles are supported (MR10.OP[7:6]==2'b01/2'b10).
  - The default T<sub>phy\_wckcsgap</sub> value covers this settings.
- WCK Postambles:
  - The 4.5\*tWCK and 6.5\*tWCK read postambles are supported (MR10.OP[3:2]==2'b01/2'b10).
  - The default T<sub>phy\_wckcsgap</sub> value covers both settings.

### LPDDR4X Preamble/Postamble rules

The Tphy\_rdcsgap and Tphy\_wrcsgap DFI timing parameters provide command gap equations whenever rank switches.

- Read DQS Preamble:
  - Support for toggling 2 tCK read preamble only
    - The default T<sub>phy\_rdcsgap</sub> value covers this setting
- Read DQS Postamble:
  - Support for toggling 1.5 tCK read postamble only
    - The default T<sub>phy\_rdcsgap</sub> value covers this setting.
- Write DQS Postamble:
  - Support for all write postamble settings defined by JEDEC
  - A 1.5 tCK write postamble may be required by certain DRAM implementations, refer to DRAM datasheets if this is required. In case 1.5 tCK write postambles are enabled, W-W transaction spacing (different ranks) must increase by 1 tCK.

## 10.5 PHY Skew Limits

The following table shows the skew limits apply to the PHY:

**Table 10-7 PHY Skew Limits [LPDDR5X/5]**

Parameter	LPDDR5X/5	Unit
Write-Leveling delay <sup>4</sup>	-1.5 to + 9	tWCK
Receive-enable delay <sup>5</sup>	tWCK2DQ(min) to + 24	tWCK
Receive-enable delay <sup>5</sup> (strobeless mode)	tWCK2DQ(min) to + 8	tWCK
DQ-to-DQ Arrival time mismatch	Up to 150ps within a byte	psec
DQ flight time to DQS flight time skew <sup>2</sup>	-75 to 75	psec
CS and CA flight time to CK flight time Skew <sup>3</sup>	-150 to 150	psec
CK+DQS flight time on PCB <sup>6</sup>	0 - Up to 1000	psec

1. Refer to “DQ-to-DQ Arrival Time Mismatch” on page 315
2. Refer to “DQ Flight Time to DQS Flight Time Skew” on page 315
3. Refer to “CS, CKE, Command/Address Flight Time to CK/CK# Flight Time Skew” on page 315
4. Positive (+) number means CK path delay > DQS/WCK path delay and negative is the opposite.
5. This delay range includes both tDQSCK / tWCK2CKO and board/package delay.
6. Includes all system routing delays between PHY pin and DRAM pin

**Table 10-8** **PHY Skew Limits [LPDDR4X]**

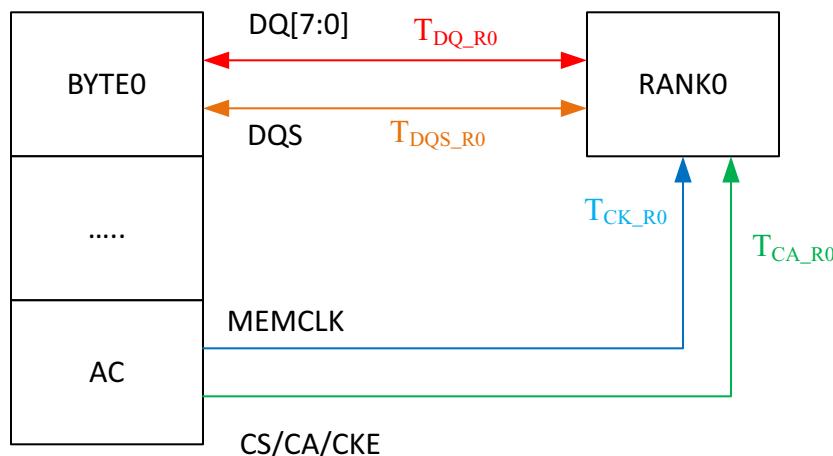
Parameter	LPDDR4X	Unit
Write-Leveling delay <sup>4</sup>	-0.5 to + 4.47	tCK
Receive-enable delay <sup>5</sup>	tDQSCK(min) to + 10.5	tCK
DQ-to-DQ arrival time mismatch	Up to 200ps within a byte	psec
DQ flight time to DQS Flight time skew <sup>2</sup>	-100 to 100	psec
CA flight time to CK flight time Skew <sup>3</sup>	-100 to 100	psec
CS flight time to CK flight time Skew <sup>3</sup>	-10 to 10 (No training for CS)	psec
CKE flight time to CK flight time Skew <sup>3</sup>	-50 to 50 (No training for CKE)	psec
CK+DQS flight time skew on PCB <sup>6</sup> (across bytes/ranks/channels)	Up to 500	psec
CK+DQS flight time on PCB <sup>6</sup>	0 - Up to 1000	psec

1. Refer to “DQ-to-DQ Arrival Time Mismatch” on page 315
2. Refer to “DQ Flight Time to DQS Flight Time Skew” on page 315
3. Refer to “CS, CKE, Command/ Address Flight Time to CK/CK# Flight Time Skew” on page 315
4. Positive (+) number means CK path delay > DQSpah delay and negative is the opposite.
5. This delay range includes both tDQSCK / tDQS2DQ and board/package delay.
6. Includes all system routing delays between PHY pin and DRAM pin

The following sections describe the table footnotes.

## 10.5.1 Skew Parameters

**Figure 10-2 Skew Parameters**



## 10.5.2 Write Leveling Delay

- Measured at DRAM DQS and CK pin
- PHY LCDL will be trained to compensate for this.
- Includes both board/package flight time and DRAM tDQSS.
  - TCK\_R0 - TDQS\_R0 for Rank 0

## 10.5.3 Receive-enable Delay

- Measured at reads at phy pin.
- PHY LCDL will be trained to compensate for this.
- For LPDDR5X/5, includes both board/package flight time and DRAM tWCK2DQO
  - TCK\_R0 + TDQS\_R0 + Twck2dqo for Rank 0 (LPDDR5)
  - Receive-enable, the total supported range is given in [Table 10-2](#) on page 304. However, the total round trip delay difference on CK and DQS path on PCB and DRAM between farthest RANK/DBYTE and nearest RANK/BYTE across channel must be less than 11 tWCK. This is due to the finite depth of the receive data\_fifo and the requirement for the PHY to send fully aligned receive data to the memory controller.
  - Example System 1: trained RxEn (across bytes, ranks) range = {10 : 20} tWCK
  - Example System 2: trained RxEn (across bytes, ranks) range = {7 : 18} tWCK
  - Note, it is recommended to limit CK and DQS path round trip delay including PCB and DRAM for farthest RANK/DBYTE to be less than 20 tWCK for system to work at highest speed
- For LPDDR4X, includes both board/package flight time and DRAM tDQSCK
  - TCK\_R0 + TDQS\_R0 + Tdqsck for Rank 0 (LPDDR4X)
  - Receive-enable, the total supported range is given in [Table 10-2](#) on page 304. However, the total round trip delay difference on CK and DQS path on PCB and DRAM between farthest RANK/DBYTE and nearest RANK/BYTE across channel must be less than 2 MEMCLK. This is

due to the finite depth of the receive data\_fifo and the requirement for the PHY to send fully aligned receive data to the memory controller.

- ❑ Example System 1: trained RxEn (across bytes, ranks) range = {4 : 10} MEMCLK
- ❑ Example System 2: trained RxEn (across bytes, ranks) range = {5 : 9} MEMCLK
- ❑ Note, it is recommended to limit CK and DQS path round trip delay including PCB and DRAM for farthest RANK/DBYTE to be less than 10 MEMCLK for system to work at highest speed

#### 10.5.4 DQ-to-DQ Arrival Time Mismatch

- Measured on reads at phy pins. PHY LCDLs will be trained to compensate for this delay.
- Timing budget shall add up all system terms and account for any difference to the supported range
- Measured on writes at DRAM PIN assuming no skew at phy pin.
- Does not include ISI (clock pattern)



**Note** It is strongly recommended to keep all flight times across the DQ's matched within 10ps for signal integrity reasons. It is responsibility of the system designer to mitigate signal integrity issues arising from cross talk alignment due to flight time mismatch.

#### 10.5.5 DQ Flight Time to DQS Flight Time Skew

- Includes all system routing delays from PHY pin to DRAM pin
- Includes all DRAM delays, including tDQSQ.
  - ❑ Difference in TDQ\_R0[n] and TDQS\_R0
- Timing budget shall add up all system terms and account for any difference to the supported range

#### 10.5.6 CS, CKE, Command/Address Flight Time to CK/CK# Flight Time Skew

- Includes all system routing delays to the DRAM ball
- Includes all DRAM delays
  - ❑ Difference in TCA\_R0[n] and TCK\_R0
- Timing budget shall add up all system terms and account for any difference to the supported range

## 10.6 DFI Low Power Interface

The DFI Low Power Interface can be used to perform clock gating when there is a sufficient period of time between traffic through the PHY.

Entering and exiting the DFI LP state is performed according to the DFI LP protocol and may be performed while the dfi interface is active (for example, dfi\_init\_complete asserted and no ongoing frequency change handshake).

Prior to performing frequency or power state changes (including LP2/LP3), the dfi\_lp\_{data,ctrl}\_req must be deasserted, since this interface must be inactive prior to frequency changes through dfi\_init\_start. This clean exit from DFI LP prior to frequency or power state changes ensures safe and well defined transitions and behaviors between power-states, while also preventing possible violations of the dfi\_lp handshakes.



**Note** The LP2/LP3 power states provides superior power savings to that of the DFI LP alone.

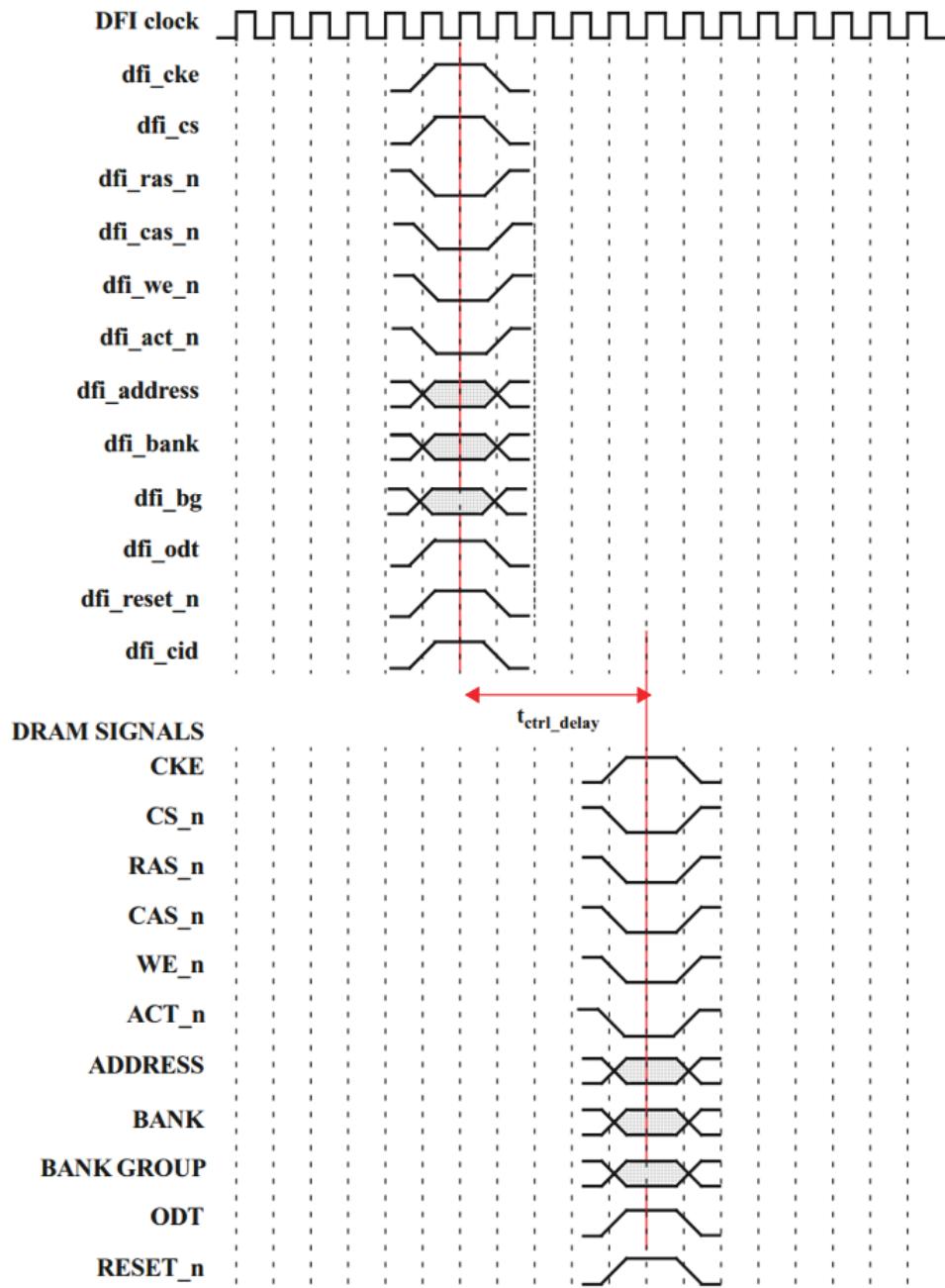
In two-channel operation, the DFI Low Power Interfaces operate independently of one another based on csrDfiMode configuration.

## 10.6.1 Entering DFI LP State

### 10.6.1.1 Entering DFI LP CTRL

Controller can assert dfi\_lp\_ctrl\_req at the end of tctrl\_delay.

**Figure 10-3 Tctrl\_delay**

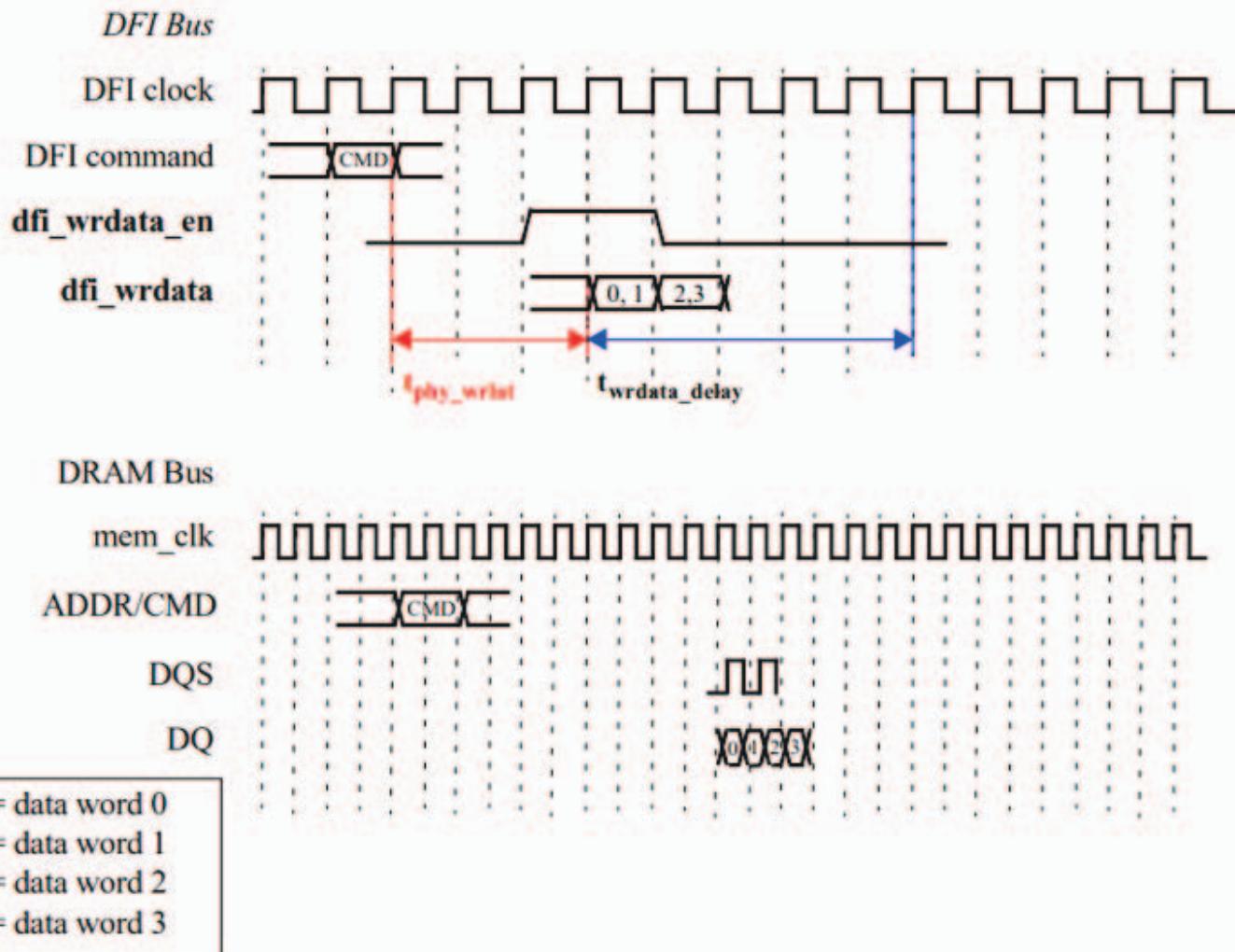


## 10.6.1.2 Entering DFI LP DATA

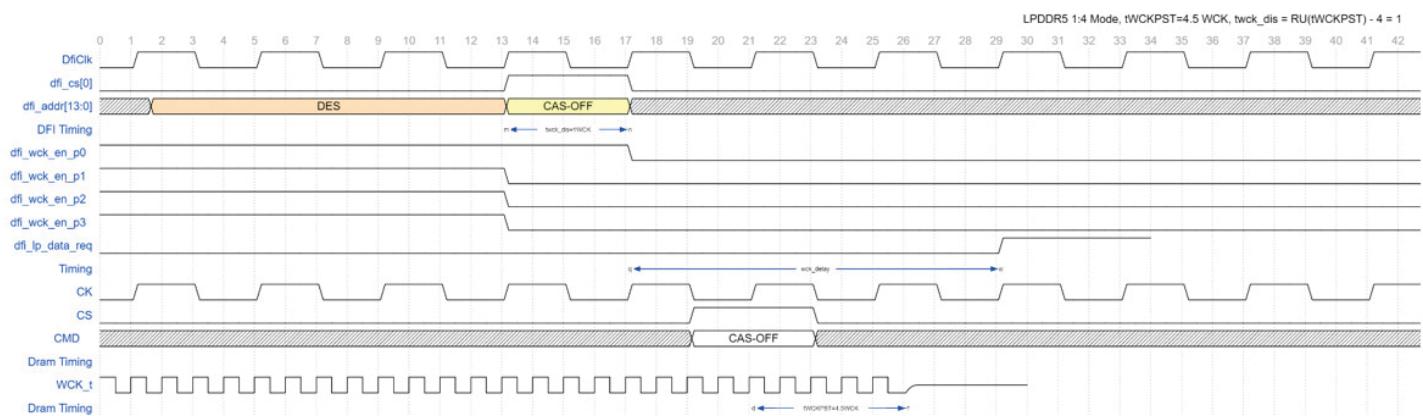
### 10.6.1.2.1 Last Command/Traffic = WRITE, Before Entering DFI LP DATA

In LPDDR4X mode, Controller can assert dfi\_lp\_data\_req at end of tphy\_wrdata\_delay, as per below diagram in DFI 5.0 specification.

Figure 10-4 Twrdata\_delay

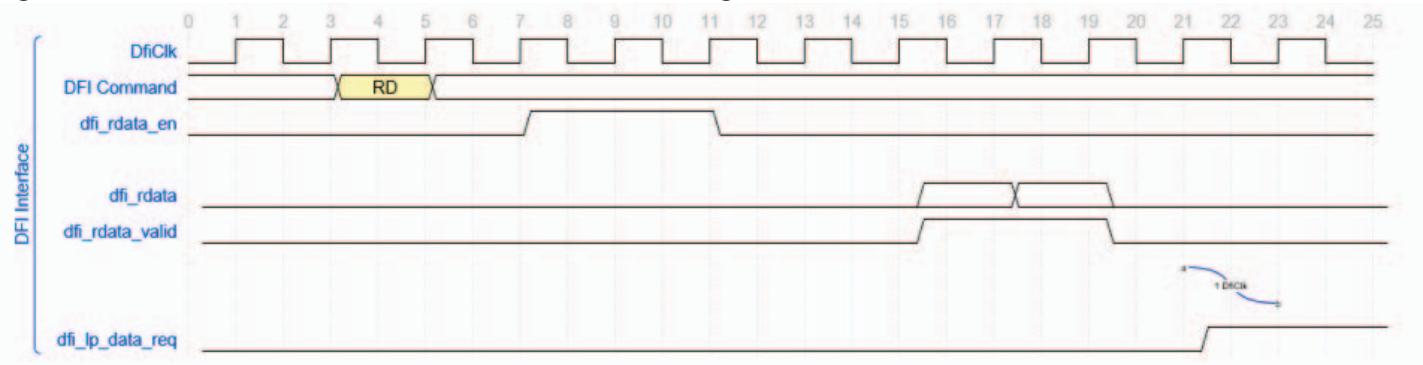


In LPDDR5 mode, Controller can assert dfi\_lp\_data\_req at end of twck\_delay, as per below timing diagram.  
Note: twck\_delay parameter is not defined in DFI specification.

**Figure 10-5 LPDDR5 WCK Free Run Mode**

### 10.6.1.2.2 Last Command/Traffic = READ, Before Entering DFI LP DATA

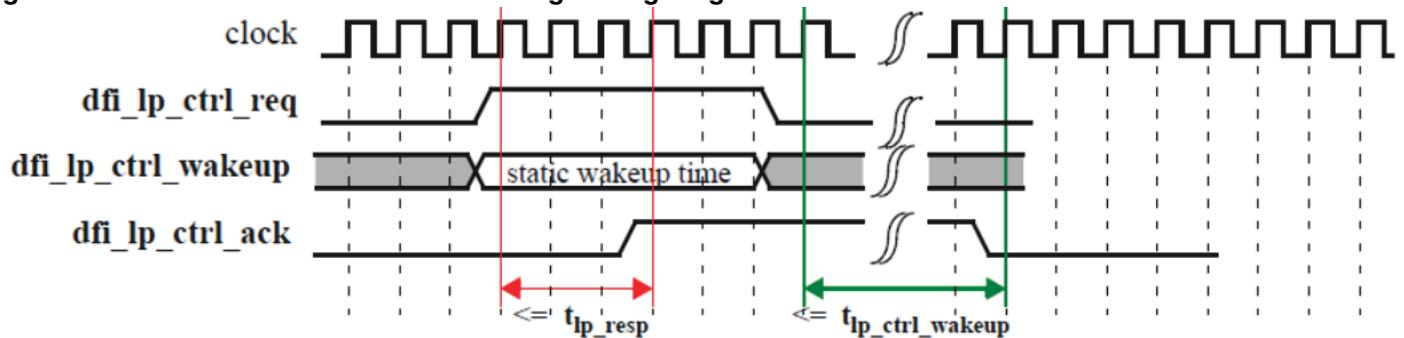
Controller can assert *dfi\_lp\_data\_req* at 1 DfiClk cycle after the last return READ Data Valid, as per below diagram below:

**Figure 10-6 Last Return READ Data Valid Before Entering DFI LP State**

## 10.6.2 Exiting DFI LP State

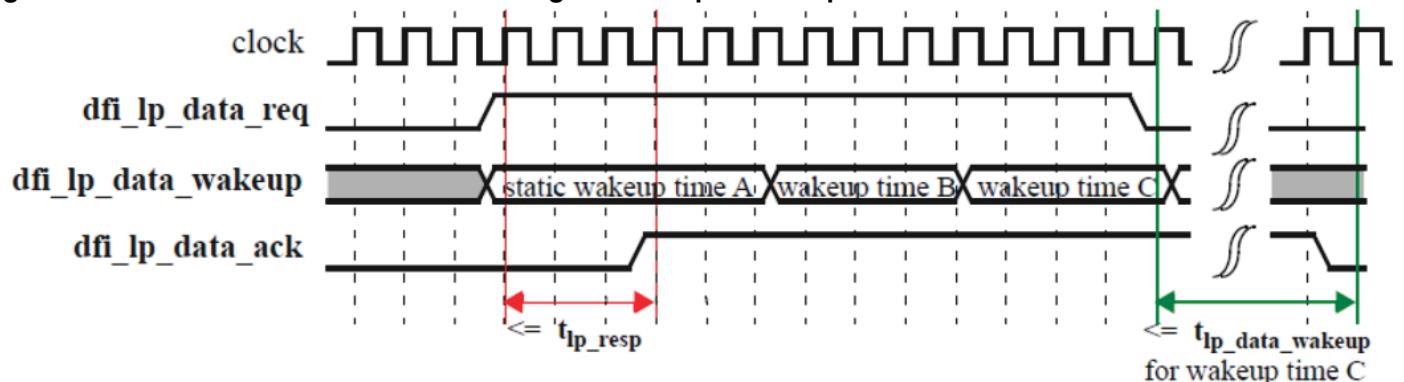
### 10.6.2.1 Exiting DFI LP CTRL

Controller can assert DFI AC traffic at end of *Tlp\_ctrl\_wakeup*, as per below diagram in DFI 5.0 specification.

**Figure 10-7 Low Power Control Handshaking Timing Diagram**

### 10.6.2.2 Exiting DFI LP DATA

Controller can assert DFI write or read data traffic ( dfi\_wrdata\_en, dfi\_wck\_en, or dfi\_rddata\_en) at the end of Tlp\_data\_wakeup, as per below diagram in DFI 5.0 specification.

**Figure 10-8 Low Power Control Handshaking with Multiple Wakeup Times**

### 10.6.3 DFI LP Data Interface

PHY supports following power saving modes:

**Table 10-9 Power Saving Modes when Dfi\_lp\_data\_req Asserts**

		No Power Saving	Maximum Power Savings <sup>a</sup>
Scenarios		1. csrDfiLPDataEn=0 OR 2. csrDfiLPDataEn=1 and dfi_lp_data_wakeup=0	csrDfiLPDataEn=1 && dfi_lp_data_wakeup >0

		No Power Saving	Maximum Power Savings <sup>a</sup>
<b>PHY Actions</b>	PUB acknowledges the dfi_lp_data_req input.	Yes	Yes
	PUB _deasserts_ DfiClkEn inputs to Dx4 and Dx5 HM	No	Yes <sup>1</sup>
	PUB _deasserts_ DfiClkEn inputs to PUB_DX	No	Yes
	PUB stops the Pclk_Dq input to all Dx4/5 HM from PAC HM.	No	Yes
	PUB stops the TxClk in all Dx4 and Dx5 HM	Yes	Yes
	Wakeup Time in DfiClk	$1 + 2^*MISC^b + csrLpDataAckDelay$	$2 + 2^*MISC^b + csrLpDataAckDelay$

- a. For very short assertion (< 4 DfiClk) of dfi\_lp\_data\_req, power-saving measures will not be activated.  
b. MISC is the setting of DWC\_LPDDR5XPHY\_PIPE\_DFI\_MISC.



**Note** dfi\_lp\_data\_req may be entered anytime there are no in-progress dram read/write transactions.

## 10.6.4 DFI LP Control Interface

PHY supports following three power saving modes:

**Table 10-10 Power Saving Modes when Dfi\_lp\_ctrl\_req Asserts**

		No Power Saving	Fast-Wakeup Power Savings <sup>3</sup>	Maximum Power Savings <sup>3</sup>
Scenarios		1. csrDfiLPCtrlEn=0 OR 2. csrDfiLPCtrlEn=1 and dfi_lp_ctrl_wakeup=0	csrDfiLPCtrlEn=1 and dfi_lp_ctrl_wakeup=1	csrDfiLPCtrlEn=1 && dfi_lp_ctrl_wakeup >1
PHY Actions	PUB acknowledges the dfi_lp_ctrl_req input.	Yes	Yes	Yes
	PUB _deasserts_DfiClkEn input to CKX2 and ACX2 HMs	No	No	No : dfi_dram_clk_disable <sup>4</sup> is low Yes : dfi_dram_clk_disable <sup>4</sup> is high
	PUB stops the Pclk_Ca input to CKX2 and ACX2 HMs from PAC HM.	No	No	No : dfi_dram_clk_disable <sup>4</sup> is low Yes : dfi_dram_clk_disable <sup>4</sup> is high
	PUB gates the TxClk Clock in non-CK HM using TxClkEn.	No	Yes <sup>3</sup>	Yes <sup>3</sup>
	PHY MemClk output state <sup>2</sup>	depends on dfi_dram_clk_disable <sup>3</sup> state	depends on dfi_dram_clk_disable state	depends on dfi_dram_clk_disable <sup>4</sup> state
	Wakeup Time in DfiClk	1 + 2*MISC <sup>5</sup> + csrLpCtrlAckDelay	2 + 2*MISC <sup>5</sup> + csrLpCtrlAckDelay	4 + 2*MISC <sup>5</sup> + csrLpCtrlAckDelay

Table Notes:

1. The MEMCLK outputs to the dram will be running unless the dfi\_dram\_clk\_disable pins are asserted-for additional power-savings)
2. The dfi\_dram\_clk\_disable should not change while dfi\_lp\_ctrl\_req=1 or dfi\_lp\_ctrl\_ack=1.
3. For very short assertion (< 5 DfiClk) of dfi\_lp\_ctrl\_req, some power-saving measures may not be activated.
4. In LPDDR4X mode, every bit of dfi\_dram\_clk\_disable must be high to stop MEMCLK output
5. MISC: Setting of DWC\_LPDDR5XPHY\_PIPE\_DFI\_MISC

## 10.7 DFI Update Interface

As per the DFI spec, the DFI update interface is intended to facilitate various operations that require interruption of transmission on the DFI interface. The update can be initiated by either the Controller (using the dfi\_ctrlupd\_req signal) or the PHY (using the dfi\_phyupd\_req signal).



**Note** In two-channel operation, the PHY Update interfaces operate independently of one another.

### 10.7.1 Controller initiated updates

The Controller initiated update interface consists of the following signals:

**Table 10-11 Controller Initiated Update Interface Signals**

Signal Name	Source	Width	Description
dfi_ctrlupd_req	Controller	1	This is a Controller initiated update. Assertion of this signal indicates that the DFI interface will be idle for enough time for the PHY to perform an update. If the PHY acknowledges the request by asserting dfi_ctrlupd_ack, then the controller must continue to hold this signal high until the PHY de-asserts dfi_ctrlupd_ack.
dfi_ctrlupd_ack	PHY	1	This is the acknowledge signal to the Controller initiated update. Once the PHY asserts this, it must be de-asserted before tctrlupd_max expires.
dfi_ctrlupd_type	Controller	2	Specified the type of update requested by the controller. 2'h0 specifies to perform PHY delay element compensation.

The PHY uses the controller initiated updates to perform periodic update to Delay Element and IO driver impedance and ODT due to temperature and voltage drift.

Note 1: PHY does not assert dfi\_ctrlupd\_ack (and does not VT compensate the delay elements) when dfi\_ctrlupd\_req is asserted within 24 cycles of the deassertion of dfi\_ctrlupd\_req if it was successfully acknowledged.

Note 2: MemClk LCDL is not VT updated.

## 10.7.2 PHY Initiated Updates

The PHY update interface consists of the following signals:

**Table 10-12 PHY Update Interface Signals**

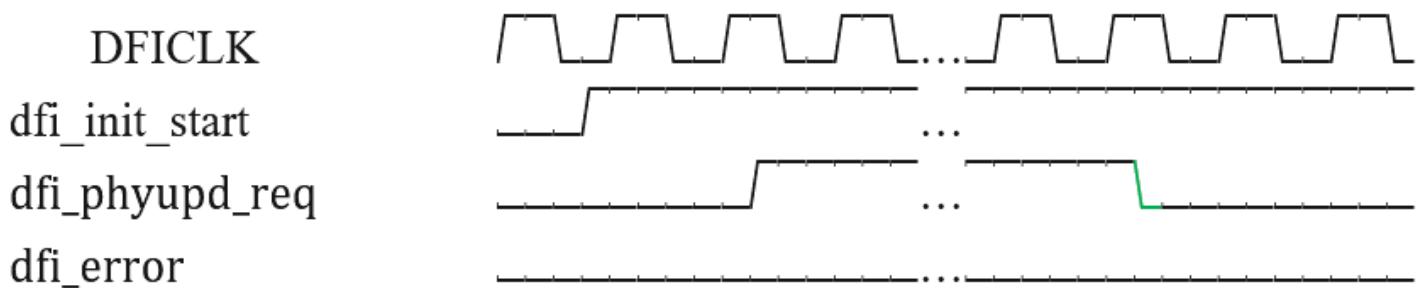
Signal Name	Source	Width	Description
dfi_phyupd_req	PHY	1	This is a PHY initiated update request. Assertion of this signal indicates that the PHY is requesting that the Controller hold DFI interface idle for enough time for the PHY to perform an update. The Controller must assert the dfi_phyupd_ack signal within tphyupd_resp cycles of the assertion of this signal. When the Controller acknowledges the request by asserting dfi_phyupd_ack, then the Controller must continue to hold dfi_phyupd_ack high until the PHY de-asserts dfi_phyupd_req.
dfi_phyupd_type[1:0]	PHY	2	This is the PHY initiated update select signal. The value of this bus indicates what type of update the PHY is requesting and determines the timing parameters for the update; this PHY will only employ tphyupd_type0 (2'b00). This value will remain constant the entire time the dfi_phyupd_req is asserted.
dfi_phyupd_ack	Controller	1	This is the acknowledge signal from the Controller in response to the PHY initiated update request.

The PHY only initiates an update for the purposes of performing the same Delay Element Compensation that it does during the Controller initiated updates. Internally, the PHY has a counter to guarantee that the Delay Element Compensation occurs often enough for the temperature and voltage drift will not adversely affect the memory operation. The counter value is defined by the value in the DFIPHYPDCNT CSR.

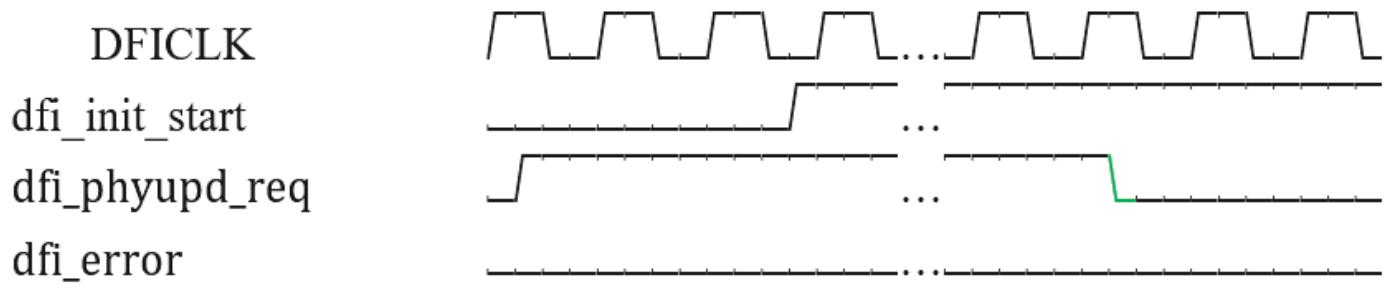
If the counter expires, it will issue an update request via the dfi\_phyupd\_req signal with the dfi\_phyupd\_type[1:0] field set to 2'b00. (All other values of the type field are unused).

If the Controller acknowledges the request, the PHY will perform the compensation and reset the counter. If the Controller does not acknowledge the request, the PHY will assert the dfi\_error signal (with dfi\_error\_info set to 4'b0000). This indicates that the PHY is beyond its compensation time, and correct memory operation is no longer guaranteed.

In the event that dfi\_init\_start is asserted, the state of the PHY update interface can safely be ignored. If the dfi\_init\_start is asserted prior to the controller observing the dfi\_phyupd\_req, the PHY will de-assert the dfi\_phyupd\_req prior to the frequency change or initialization being completed. The de-assertion will occur within 156 cycles of the assertion of dfi\_init\_start. In multi-channel operation, this is measured from the latest dfi\_init\_start assertion. This is because the PHY synchronizes these events, as described in section “[Multi-channel Operation](#)” on page [330](#).

**Figure 10-9 dfi\_init\_start prior to dfi\_phyupd\_req**

If the dfi\_init\_start is asserted after the dfi\_phyupd\_req is asserted, the PHY may assert the dfi\_error signal if the tphyupd\_resp time has been violated. The error signal can be safely ignored in this case.

**Figure 10-10 dfi\_init\_start after dfi\_phyupd\_req**

**Note** In order to ensure continual proper operation of the memory system, it is recommended the dfi\_phyupd\_ack be deasserted by the memory controller within 8 Dfi clock cycles of the deassertion of dfi\_phyupd\_req.

### 10.7.3 PHY Update Interface Timing Parameters

Refer to “[PUB Parameters](#)” on page [128](#) for PHY timing parameters for the update interface requests.

### 10.7.4 Collisions Between PHY Update and DFI LP

The intention of the dfi\_error behavior is to provide a complete solution that prevents lock-up. If the MC does not violate tphyupd\_resp, then the assertion of the error should not be an issue.

MC shouldn't assert DFI LP request if PHY Update request has been already ACKed. If MC receives PHY Update request while DFI LP request is already ACKed, MC should de-assert LP request, wait for de-assertion of LP ACK and then ACK the PHY Update request.

The dfi\_phyupd\_req will be deasserted within 156 cycles of the dfi\_init\_start assertion. The purpose of this time is to ensure that there is flexibility in the pipelining of the interfaces. The update and frequency change interfaces are not intended to be low latency and high performance interfaces.

## 10.8 DFI Error Interface

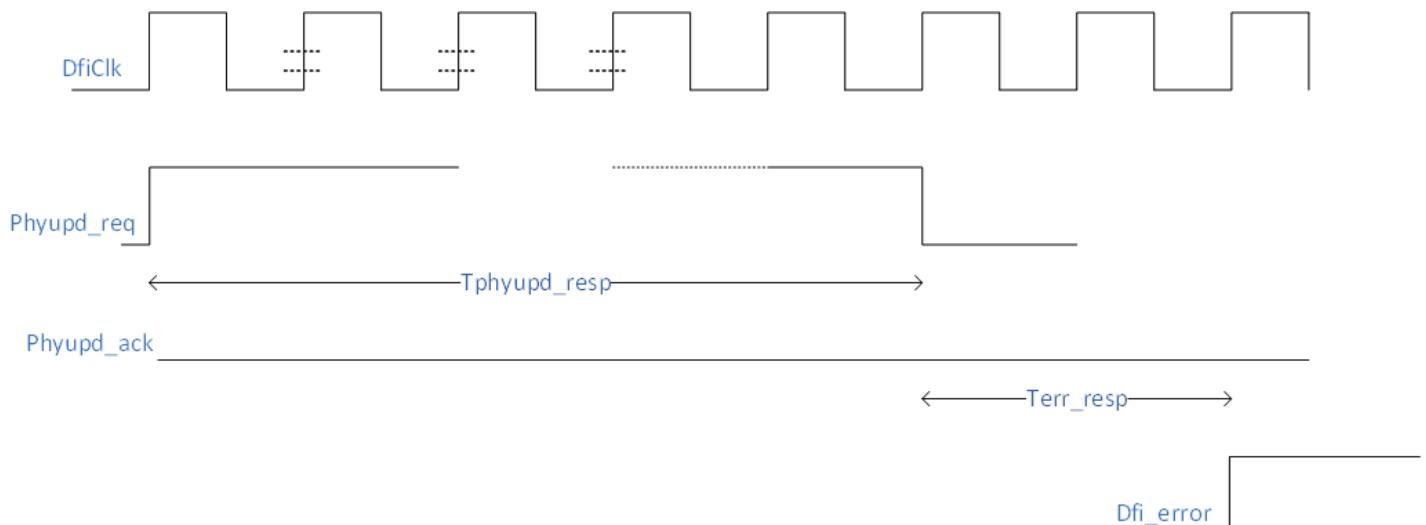
As per DFI specification, the error interface is used to convey error conditions to MC. PHY will assert dfi\_error signal with dfi\_error\_info[3:0], within Terr\_resp cycles after error condition occurs.

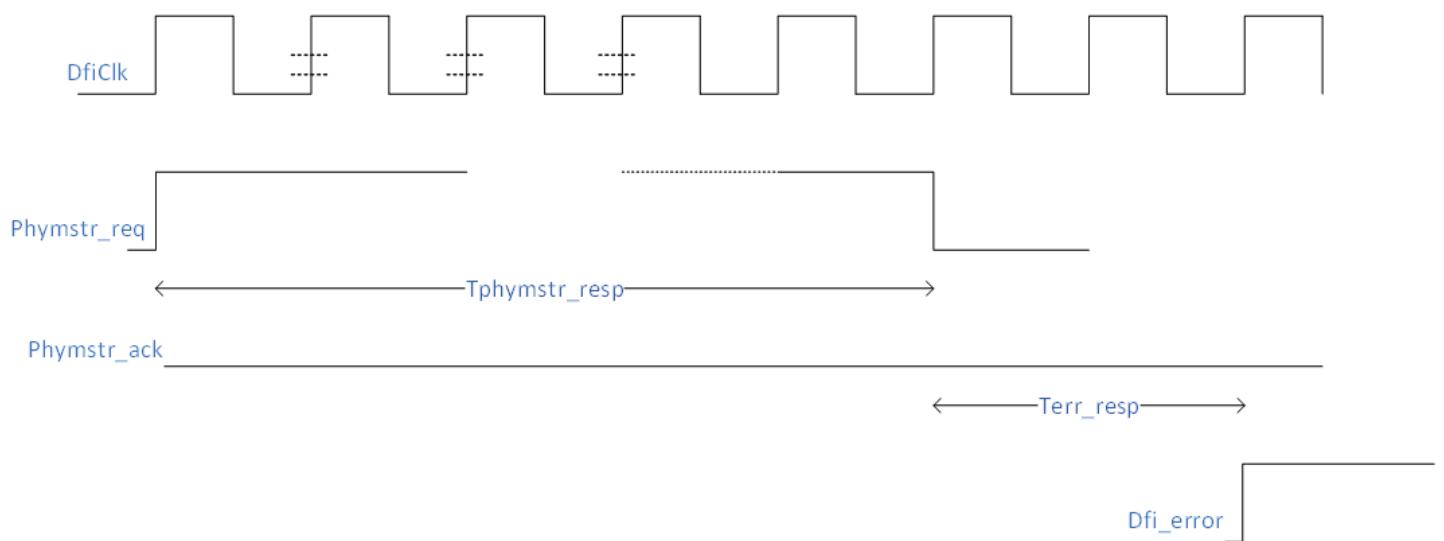
PHY asserts dfi\_error in following conditions:

**Table 10-13 DFI Error Codes**

Error Condition	dfi_error_info[3:0]
MC does not assert dfi_phymstr_ack within csrPhyMstrMaxReqtoAck cycles after PHY asserted dfi_phymstr_req	4'h1
MC does not assert dfi_phyupd_ack within csrDFIPHYUPDRESP cycles after PHY asserted dfi_phyupd_req	4'h0

**Figure 10-11 Terr\_resp(PHYUPD)**



**Figure 10-12 Terr\_resp(PMI)**

## 10.9 DFI Master Interface

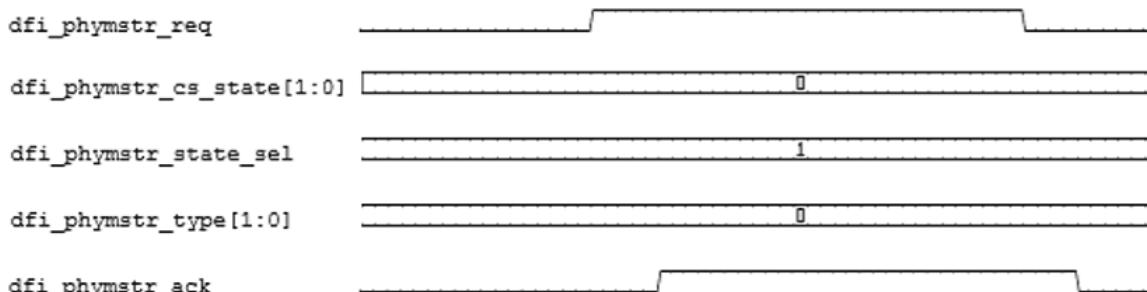
As per the DFI specification the DFI PHY Master interface (PMI) is intended to facilitate various operations that require interruption of transmission on the DFI interface. The PMI is used to initiate the Dram Drift Compensation operations. The PMI uses internal timers to determine when the DRAM needs to be retrained. These timers are set up during the initialization process and are activated by frequency change operations on the DFI interface. There is a separate timer for each PState that is used when that PState is active. When the timer counts down to zero, it will request control of the DRAM by asserting the `dfi_phymstr_req` signal. Once the request has been acknowledged, it will perform the DDC operation and return the control of the DRAM to the memory controller automatically.



**Note** For more information on the operation of the PMI and the requirements, refer to the DFI specification.

The DFI PHY Master Interface operation can be seen in the following waveform:

**Figure 10-13 DFI PHY Master Interface Operation**



When the `dfi_phymstr_req` signal is asserted, the other PHY master signals will always be set as:

- `dfi_phymstr_cs_state[1:0]` will always be set to 0
- `dfi_phymstr_state_sel` will always be set to 1
- `dfi_phymstr_type[1:0]` will always be set to 0

If the PHY Master Interface is enabled and the `dfi_phymstr_ack` signal is not asserted within `tphymstr_resp` cycles after the `dfi_phymstr_req` signal is asserted, the `dfi_error` signal will be asserted (with `dfi_error_info` set to 4'b0001).

### 10.9.1 Two-channel Operation

Two-channel operation: Even though there are separate `dfi0_phymstr_*` and `dfi1_phymstr_*` signals, a DDC operation affects the entire PHY. Thus, after `dfi0_phymstr_req` and `dfi1_phymstr_req` are asserted, both MC channels must assert their `dfi*phymstr_ack` signals within the required interval to initiate a DDC. If either channel does not assert `phymstr_ack` within the interval, an error condition occurs.

## 10.10 PHY DFI Sideband Interactions

The following section describes the expected behavior of the PHY and Memory Controller.

### 10.10.1 Supported DFI Sideband Interfaces

The PHY supports the following DFI Sideband interfaces [for full details on each interface, refer to the appropriate section of the DFI Specification].

- DFI Frequency change Interface
  - This interface is used by the Memory Controller to Start the DFI interface at boot, or to request a frequency change
- Relevant Signals
  - dfi0\_init\_start, dfi0\_init\_complete, dfi0\_frequency[4:0], dfi0\_freq\_ratio[1:0], dfi0\_freq\_fsp[1:0]
  - dfi1\_init\_start, dfi1\_init\_complete, dfi1\_frequency[4:0], dfi1\_freq\_ratio[1:0], dfi1\_freq\_fsp[1:0]
- DFI Low Power Interface
 

This interface is used by the Memory Controller to notify the PHY of extended traffic idle times that may also be used for entering PHY low power states

  - Relevant Signals
    - dfi0\_lp\_ctrl\_req, dfi0\_lp\_ctrl\_wakeup[4:0], dfi0\_lp\_ctrl\_ack
    - dfi0\_lp\_data\_req, dfi0\_lp\_data\_wakeup[4:0], dfi0\_lp\_data\_ack
    - dfi1\_lp\_ctrl\_req, dfi1\_lp\_ctrl\_wakeup[4:0], dfi1\_lp\_ctrl\_ack
    - dfi1\_lp\_data\_req, dfi1\_lp\_data\_wakeup[4:0], dfi1\_lp\_data\_ack
- DFI Controller Update Interface
  - This interface is used by the Memory Controller to notify the PHY of traffic idle times. The PHY uses these intervals to enable VT compensation updates for internal PHY delay lines and to reset the read-data fifo pointers.
  - Relevant Signals
    - dfi0\_ctrlupd\_req, dfi0\_ctrlupd\_type[1:0], dfi0\_ctrlupd\_ack
    - dfi1\_ctrlupd\_req, dfi1\_ctrlupd\_type[1:0], dfi1\_ctrlupd\_ack
- DFI PHY Update Interface
  - This interface is used by the PHY to request traffic idle times to enable VT compensation updates for internal PHY delay lines
  - Relevant Signals
    - dfi0\_phyupd\_req, dfi0\_phyupd\_type[1:0], dfi0\_phyupd\_ack
    - dfi1\_phyupd\_req, dfi1\_phyupd\_type[1:0], dfi1\_phyupd\_ack
- DFI PHY Master Interface
  - This interface is used by the PHY to request complete control of the memory bus to enable sequences to compensate for DRAM drift [for example, drift in tDQSCK and tDQS2DQ]
  - Relevant Signals

- dfi0\_phymstr\_req, dfi0\_phymstr\_type[1:0], dfi0\_phymstr\_state\_sel, dfi0\_phymstr\_cs\_state[1:0], dfi0\_phymstr\_ack
- dfi1\_phymstr\_req, dfi1\_phymstr\_type[1:0], dfi1\_phymstr\_state\_sel, dfi1\_phymstr\_cs\_state[1:0], dfi1\_phymstr\_ack

### 10.10.2 Multi-channel Operation

In PHY configurations where the secondary DFI1 interface is enabled [define DWC\_LPDDR5XPHY\_NUM\_CHANNELS\_2 is defined] the DFI sidebands are also replicated (dfi1\_\* vs. dfi0\_\*).

The following interfaces are completely independent between DFI0/DFI1, and have no interaction

- DFI Low Power
- DFI Controller Update
- DFI PHY Update

The following interfaces require acknowledgment by both DFI0/DFI1 channels, and are synchronized by the PHY

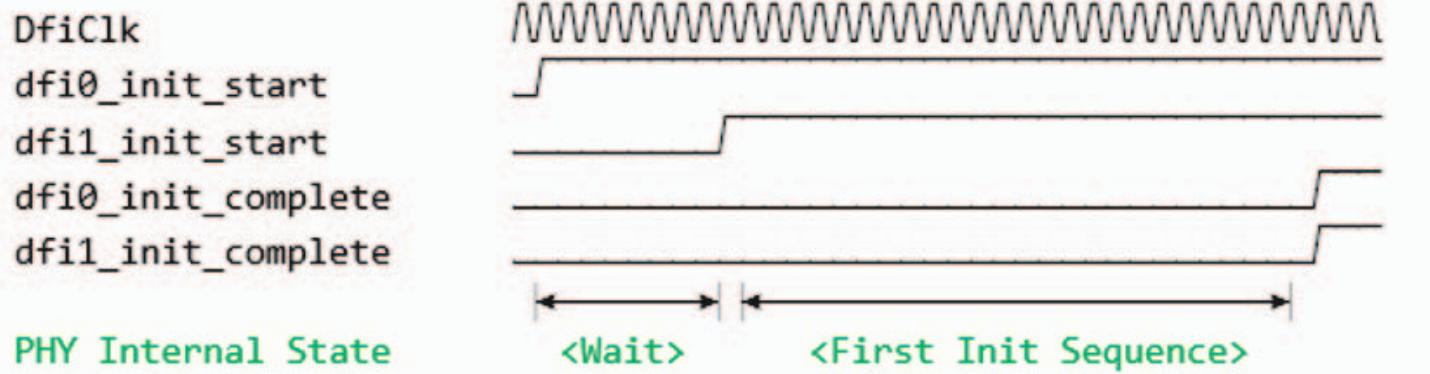
- DFI Frequency Change Interface
- DFI PHY Master Interface

#### 10.10.2.1 Synchronization of DFI Init/Frequency Change Interface

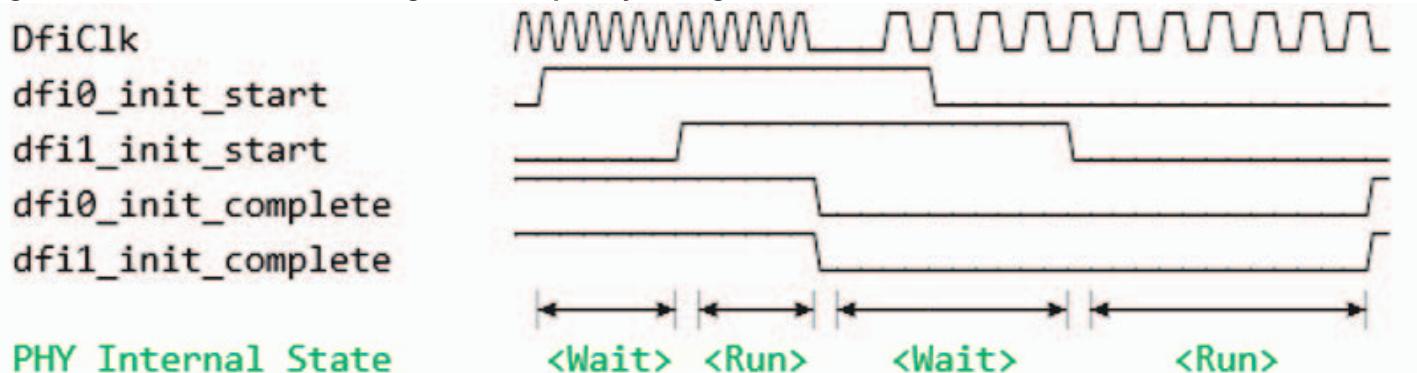
Since the PHY has a single PLL shared among both DFI channels, the first DFI Init and subsequent frequency changes must occur for both channels in concert.

- The Memory Controllers may transition the dfi{0,1}\_init\_start signals at different times and in any order.
- The PHY will wait until both dfi{0,1}\_init\_start signals are at the same state before starting the first init or frequency change.
  - The PHY will remain in the wait state until both dfi{0,1}\_init\_start signals have transitioned.
- Since the PHY will execute frequency change at the same time for both channels, the dfi{0,1}\_init\_complete signals will be asserted by the PHY simultaneously.

The following figure shows the PHY behavior during the first DFI initialization.

**Figure 10-14 PHY Behavior During the First DFI Initialization**

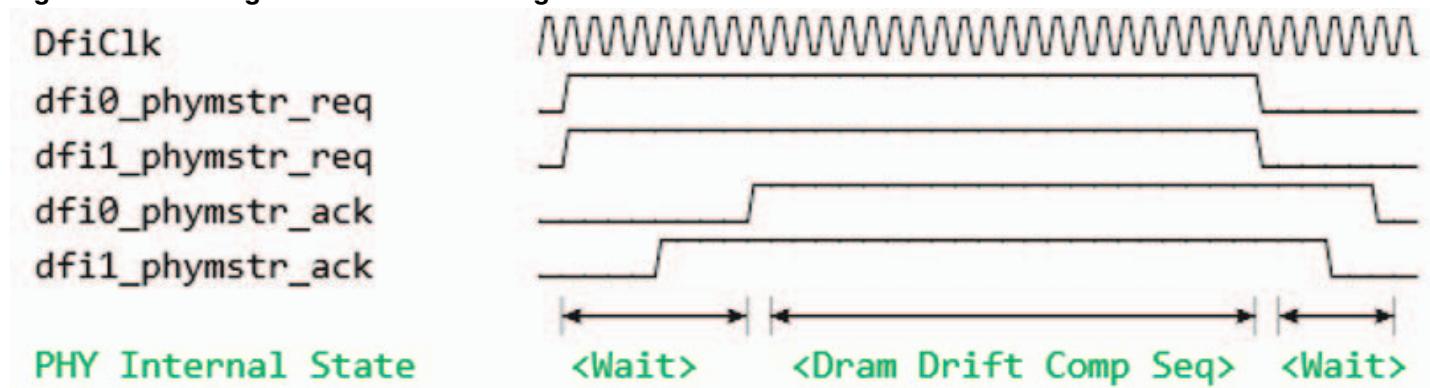
The following figure shows the PHY behavior during a DFI frequency change.

**Figure 10-15 PHY Behavior During a DFI Frequency Change**

### 10.10.2.2 Synchronization of DFI PHY Master Interface

- Since the PHY has a single dram-drift-compensation engine for both DFI channels, the DFI PHY Master events for both channels must occur in concert.
- The PHY will assert the dfi{0,1}\_phymstr\_req signals simultaneously
- The Memory Controllers may acknowledge by asserting the dfi{0,1}\_phymstr\_ack signals in any order.
- The PHY will wait until both dfi{0,1}\_phymstr\_ack signals are asserted. When both acknowledgements are received by the PHY, the internal dram-drift-compensation sequence will start.
- After the dram-drift compensation sequence is complete, the PHY will de-assert both dfi{0,1}\_phymstr\_req signals in the same cycle.
- The Memory controllers may de-assert their respective dfi{0,1}\_phymstr\_ack signals in any order.
- If the Memory Controller does not acknowledge a dfi{0,1}\_phymstr\_req within the window programmed in csrPhyMstrMaxReqToAck, dfi{0,1}\_error is asserted, and 0x1 is driven on dfi{0,1}\_error\_info.

The following figure shows the PHY behavior during a DFI PHY Master Transaction.

**Figure 10-16 Change PHY Behavior during a DFI PHY Master Transaction**

## 10.11 DFI Sideband Collisions

### 10.11.1 DFI Sideband Simultaneous Collisions

Since the DFI sideband requesters are a mix of the PHY and MC, there is the possibility of collisions between the interfaces.

The following sidebands are initiated by the Memory Controller, and should not be requested simultaneously. That is, the asserting edge of the requests should not occur in the same DfiClk cycle.

- DFI Init/Frequency Change
- DFI Controller Update
- DFI Low Power

The following sidebands are initiated by the PHY. The PHY logic ensures that the following are not asserted in the same DfiClk cycle.

- DFI PHY Update
- DFI PHY Master

The following table describes the expected behavior of the PHY and MC in case of simultaneous collisions of the sidebands.

**Table 10-14 Expected Behavior of the PHY and MC in Case of Simultaneous Collisions of the Sidebands**

	DFI Init / Frequency Change	DFI Controller Update	DFI Low Power	DFI PHY Update Request	DFI PHY MASTER Request
DFI Init / Freq Change	-	Illegal/Undefined. MC should not do this	Illegal/Undefined. MC should not do this	PHY starts the Freq. Change sequence. PHY de-asserts dfi_phyupd_req	PHY starts the Freq. Change sequence. PHY de-asserts dfi_phymstr_req
	-				
	-				
	-				
	-				
	-				
DFI Controller Update	-	-	Illegal/Undefined. MC should not do this	PHY asserts dfi_ctrlupd_ack. MC should acknowledge PHY MASTER request after DFI Controller Update is complete	PHY asserts dfi_ctrlupd_ack. MC should acknowledge PHY MASTER request after DFI Controller Update is complete
	-	-			
	-	-			
	-	-			
	-	-			
	-	-			

	<b>DFI Init / Frequency Change</b>	<b>DFI Controller Update</b>	<b>DFI Low Power</b>	<b>DFI PHY Update Request</b>	<b>DFI PHY MASTER Request</b>
DFI Low Power	-	-	-	PHY asserts dfi_lp_{ctrlldata}_ack.	PHY asserts dfi_lp_{ctrlldata}_ack.
	-	-	-	MC should acknowledge PHY Update request after DFI Low Power transaction is complete	MC should acknowledge PHY MASTER request after DFI Low Power transaction is complete
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
DFI PHY Update Request	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	PHY logic guarantees this cannot occur .
DFI PHY MASTER Request	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	-	-
	-	-	-	PHY logic guarantees this cannot occur	-



**Note** dfi\_\*\_{ack,req} corresponds to the dfi{0,1}\_\*\_{ack,req} as appropriate  
Simultaneous refers to the setup of the signals to the internal PHY DfiClk.

### 10.11.2 DFI Sideband Sequential Collisions

Some sequential overlap of the sidebands is permitted/undefined by the DFI specification.

The following table describes the expected behavior of the PHY and MC in case of simultaneous collisions of the sidebands.

**Table 10-15 Expected Behavior of the PHY and MC in Case of Sequential Collisions of the Sidebands<sup>1</sup>**

<b>1st EVENT</b>	<b>2nd EVENT</b>	<b>DFI Init / Freq Change</b>	<b>DFI Controller Update</b>	<b>DFI Low Power</b>	<b>DFI PHY Update Request</b>	<b>DFI PHY MASTER Request</b>
DFI Init / Freq Change	-	Illegal/Undefined. MC should not do this	Illegal/Undefined. MC should not do this	PHY starts the Freq. Change sequence. PHY de-asserts dfi_phyupd_req	PHY starts the Freq. Change sequence. PHY de-asserts dfi_phymstr_req	
	-					
	-					
	-					
	-					
	-					
DFI Controller Update	Illegal/Undefined. MC should complete DFI_ctrlupd transaction before requesting Freq. Change	-	-	PHY asserts dfi_ctrlupd_ack.  if csrDxOutPipeEn = zero , PHY will not generate dfi phyupd request. if csrDxOutPipeEn = non-zero, PHY may or may not generate dfi phyupd request. If it generates, it will de-assert within a few cycles.	PHY asserts dfi_ctrlupd_ack.  if csrDxOutPipeEn = zero , PHY will not generate dfi phyupd request. if csrDxOutPipeEn = non-zero, PHY may or may not generate dfi phyupd request. If it generates, it will de-assert within a few cycles.	PHY asserts dfi_ctrlupd_ack.  MC should acknowledge PHY MASTER request after dfi_ctrlupd transaction is complete
DFI Low Power	Illegal/Undefined. MC should complete DFI_LP transaction before requesting Freq. Change	PHY asserts dfi_lp_{ctrlldata}_ack  PHY ignores dfi_ctrlupd_req	-	MC should acknowledge PHY Update request only after DFI Low Power transaction is complete	PHY asserts dfi_lp_{ctrlldata}_ack.  MC should acknowledge PHY MASTER request after dfi_lp transaction is complete	

1. dfi\_\*\_{ack,req} corresponds to the dfi{0,1}\_\*{ack,req} as appropriate

**Table 10-16    Expected Behavior of the PHY and MC in Case of Sequential Collisions of the Sidebands<sup>1</sup>**  
**(continued)**

1st EVENT	2nd EVENT	DFI Init / Freq Change	DFI Controller Update	DFI Low Power	DFI PHY Update Request	DFI PHY MASTER Request
DFI PHY Update Request		If dfi_phyupd_ack=0 PHY de-asserts dfi_phyupd_req, & PHY starts Freq. Change sequence.  If dfi_phyupd_ack=1, Illegal/undefined. MC should not do this	If dfi_phyupd_ack=0 PHY de-asserts dfi_phyupd_req. & PHY asserts dfi_ctrlupd_ack.  If dfi_phyupd_ack=1, Illegal/undefined. MC should not do this	If dfi_phyupd_ack=0, PHY asserts dfi_lp_ctrl/data_ack , MC should acknowledge PHY Update request after DFI Low Power transaction is complete;  If dfi_phyupd_ack=1, Illegal/undefined. MC should not do this.	- - - - - -	PHY logic guarantees this cannot occur
DFI PHY MASTER Request		If dfi_phymstr_ack=0 PHY asserts dfi_ctrlupd_ack. MC should acknowledge PHY MASTER request after dfi_ctrlupd transaction is complete  if dfi_phymstr_ack=1 , this combination is Illegal/Undefined	If dfi_phymstr_ack=0 PHY asserts dfi_ctrlupd_ack. MC should acknowledge PHY MASTER request after dfi_ctrlupd transaction is complete  if dfi_phymstr_ack=1 , this combination is Illegal/Undefined	If dfi_phymstr_ack=0 PHY asserts dfi_lp_{ctrlldata}_ack.  MC should acknowledge PHY MASTER request after dfi_lp transaction is complete  if dfi_phymstr_ack=1 , this combination is Illegal/Undefined	PHY logic guarantees this cannot occur .	

1. dfi\_\*\_{ack,req} corresponds to the dfi{0,1}\_\*{ack,req} as appropriate

# 11

## Automotive Support

---



**Note** The following section is only applicable to products that support Automotive functionality.

---

Synopsys' Automotive IP is designed to support applications requiring ISO 26262 ASIL B and AEC Q100. This assumes that the integrity of the IP is maintained during implementation of the host SoC, Package, and PCB/System.

The following sections are about automotive safety.

Synopsys provides functional safety documentation for following standard product configuration:

- 2 Ranks
- LPDDR5X/5/4X protocols
- 2 DFI channels
- 32 data bits (16 DQ with DBI per channel)

## 11.1 LogicBIST (LBIST)

LogicBIST (LBIST) is a solution for in-system self-test of digital integrated circuits. LBIST addresses functional safety requirements set forth by standards such as ISO 26262 for the automotive semiconductor industry.

Self-test utilizes scan chains and a small amount of logic added by a DFT tool such as Synopsys DFTMax. LBIST uses pseudo-random pattern generator (PRPG) to create scan data, and a multiple-input signature register (MISR) to compare the design response to an expected value. At the end of the test, if the actual signature matches the expected signature, a PASS signal will be asserted.

Since scan patterns are generated on-chip with PRPG and test results are compressed into MISR to generate final signature for comparison, test time can be reduced because scan shift is not limited by external data. However, using pseudo random patterns might limit detection of some random resistant logic or specific pattern types.

LogicBIST is not a requirement for Synopsys Automotive PHY to meet Functional Safety ASIL targets. However, if a customer would like to implement it, details about LogicBIST implementation can be found in LBIST Application Note.

Refer to the additional LBIST Application Note for more information.

## 11.2 Anti-Valent Interrupt

PHY has following 3 interrupt sources which are safety critical.

- ACSM SRAM Parity Error
  - Connected to PHY\_TOP pins PhyInt\_fault[1:0]
- PIE SRAM Parity Error
  - Connected to PHY\_TOP pins PhyInt\_fault[3:2]
- RxFIFO Check Error
  - Connected to PHY\_TOP pins PhyInt\_fault[5:4]

Each of the above interrupts has been implemented using anti-valent flip-flops for reliability.

Each of the above interrupts has 2 bit “fault[1:0]” outputs available as PHY\_TOP pin for SOC to monitor the status.

- 2'b10 :: no fault state
- Anything else :: some fault.

Above interrupts cannot be masked. However, they have 2 CSR controls:

- Clear :: when asserted, forces the “fault[1:0]” output pins as 2'b10.
  - Refer to csrPhyInterruptClear description.
- Override :: when asserted, forces the fault[1:0] output pins as 2'b01.
  - This is for debug/testing purposes.
  - Refer to csrPhyInterruptOverride description.

**Table 11-1 Anti-Valent Interrupt**

Scenario	Inputs					Outputs	
	Reset	Interrupt Source Rising Edge	Clear	Override	fault[0]	fault[1]	
Reset	1	X	X	X	0	1	
Override	0	X	X	1	1	0	
Clear	0	0	1	0	0	1	
New Interrupt	0	1	X	0	1	0	
No Interrupt	0	0	0	0	0	1	

## 11.3 Triple Mode Redundant (TMR) Controls

Triple Mode Redundancy (TMR) insertion is to be included by the customer while implementing the PUB components of the IP for flip-flops that perform critical control functions, including enabling/disabling the microcontroller subsystem. Implementation must ensure these TMR flops are spaced apart such that an upset event does not affect more than 1 flop.

This ensures that these critical functions continue to operate as intended, even in the presence of single-bit soft errors.

The RTL module names:

- dwc\_lpddr5xphy\_csrC\_TRR\_pub.v
- dwc\_lpddr5xphy\_csrRW\_RW\_TRR\_pub.v
- dwc\_lpddr5xphy\_csr\_TRR\_pub.v

To avoid unwanted optimization on this logic, it is recommended to use the “set size\_only” attribute for the cells of the triplicated registers inside the above modules.

Examples of flops to be implemented with TMR:

- PUBMODE[0].HwtMemSrc
- MicroContMuxSel[0].MicroContMuxSel
- ContextToMicro[0].ContextToMicro
- MicroReset[3].ResetToMicro
- MicroReset[0].StallToMicro
- SequencerOverride (all bits)
- PieVecCfg (all bits)
- PieInitVecSel (all bits)
- ExternalAHBReset
- TDRDisable
- ClearPIESTallToMicro
- ForceRxDataFifoUpd
- PPTTrainSetup (all bits)

## 11.4 Scratch pad register

Each APB slave implements a 32 bit scratch pad reserved register. This CSR is not used by PHY for any functional use. Host performs write/read to this CSR periodically to test integrity of APB interface.

These registers can be identified by the “ScratchPad” name.

## 11.5 APB Read Parity

Each APB slave in the PHY implements an even parity generation logic for each 8 bits of APB\_RDATA and sends to the host using PRDATA\_PAR\_APB output.

Host re-calculate the parity with received read data (PRDATA\_APB[31:0]) and compares it with received PRDATA\_PAR\_APB[3:0] to test integrity of APB read interface.

Refer to Safety Manual for details.

## 11.6 Static State requirement

SOC must ensure that following PHY inputs remain static during functional (mission) mode:

- scan\_\*
- Reset
- Reset\_async
- BP\_PWROK
- BurnIn
- PRESETn\_APB
- WRSTN
- LBIST\_MODE
- TxBypassMode\_\*
- Iddq\_mode
- Dfi\_reset\_n

Refer to Safety Manual for details.

## 11.7 PLL Lock Indicator

PHY shall provide PLL Lock indicator, directly from PLL output without any CSR control, to PHY\_TOP pin (dwc\_ddrphy\_pll\_lock).

PLL Lock Indicator monitoring by Host is not a requirement for Synopsys Automotive PHY to meet Functional Safety ASIL targets.

SOC may choose to monitor this output as an indication of PLL coarse stability such as:

- Detection of sudden, extended problems with reference clock (>16 cycles)
- Detection of supply issues, where PLL cannot operate in intended conditions
- Detection of samples with manufacturing defects, where PLL SPO is permanently degraded

However, pll\_lock indicator cannot be used for:

- Detection of single, or few cycles, jitter events on reference clock, even though those might break the jitter budget of the PHY. Having pll\_lock=1'b does not guarantee correct operation of PHY.
- Detection of slow-occurring drifts, even those breaking the PHY specification – PLL will be able to track the slow drifts: supply, temperature and reference drifts alike.
- Detection of “bad chips” – as overall jitter budget only partially depends on PLL, it is possible to have a sample that flags pll\_lock=0'b, but otherwise operates correctly.

# 12

## Register Overview

---

This chapter describes the following:

- “Register Map Overview” on page [344](#)
- “Register Address Overview” on page [345](#)
- “Specifying Register Addresses” on page [346](#)

## 12.1 Register Map Overview

The PHY has registers that are used to configure, control or provide status of certain features of the PHY. The user accesses these registers using the configuration port write and read commands. "Register Address Map" shows the address mapping of the registers.

The following are the register rules:

- All register accesses from APB interface is in units of 32 bits, though the addressed register itself may be less than 32 bits wide.
- APB write data bits [31:16] from host should be always 0 when writing to PHY CSR.
- APB read data bits [31:16] to host should be always 0 when reading from PHY CSR.
- APB write or read from external SRAM will be always be in 32 bits.
- Values should not be written to registers that are marked reserved
- Values should not be written to undocumented addresses

The following sections describe the registers in detail. Unless specified otherwise, control register bits are assumed active-high.

## 12.2 Register Address Overview

Sets of registers are in specific register-blocks. The register-block is a group of registers which are synthesized as one, share a common Verilog module, and respond to addresses for that kind of register-block. There may be multiple instances of each kind of register-block.



- Note**
- Register addresses are 23-bits wide.
  - The architecture supports the optional feature to broadcast same value to many registers during a write operation.

Refer to the Register Address Map for all register blocks.

## 12.3 Specifying Register Addresses

The 23 bit register address is specified by the concatenation of the following fields: {PState, Block Type, Instance Number, Register}.

**Table 12-1 Definition of Register Addresses**

[22:20]	PState	Which copy of the Per-Pstate Register is active
[19:16]	Block Type	The number corresponds to the name of a physical group of register
[15:12]	Instance Number	The particular instance of the Block Type
[11:0]	Register	The particular word which is being accessed

The following sections describe the fields in detail.

### 12.3.1 PState

PState: Certain registers have 4 copies, one per PState. Each copy can be accessed individually by setting the PState field appropriately. Registers which are not copied per P-state exist at PState0.

- PState = 3'b000 : PState0
- PState = 3'b001 : PState1
- PState = 3'b010 : PState2
- PState = 3'b011 : PState3
- PState = 3'b111 : All PState

In an operational PHY, only one set of PState registers is active at a time. The PState is the operating point of the PHY (that is, the particular frequency/power operating point to which is configured).

See register PState in the primary register block for information on moving between PStates.

### 12.3.2 Block Type

Block Type: Registers are grouped together in blocks, for reasons of physical proximity and ease of replication where more than one set is needed. For example, there is one register per DBYTE and there are multiple DBYTEs. Bits [19:16] specify the block-type, according to the following table:

- Block Type = 4'h0: ACX2 / CSX2/ CKX2 Hard Macro
- Block Type = 4'h1: PUB\_DX
- Block Type = 4'h2: PUB\_PAC
- Block Type = 4'h3: PUB\_AC
- Block Type = 4'h4: TUB\_ACSM\_SRAM/TUB\_PIE\_SRAM
- Block Type = 4'h5: TUB\_ICCM/TUB\_DCCM
- Block Type = 4'h6: PAC Hard Macro
- Block Type = 4'h7: TUB\_PPGC
- Block Type = 4'h9: INITENG
- Block Type = 4'hA: ZCAL Hard Macro

- Block Type = 4'hB: PUB\_ZCAL
- Block Type = 4'hC: TUB\_DRTUB
- Block Type = 4'hD: TUB\_APBONLY
- Block Type = 4'hE: DBYTE hard macro



# 13

## Register Descriptions

This chapter details all possible registers in the IP. They are arranged hierarchically into maps and blocks (banks).

### Exists Expression

The Exist expressions (if present) indicate the combination of configuration parameters required for a register, field, or block to exist in the memory map. The expression is only valid in the local context and does not indicate the conditions for existence of the parent. For example, the Exists expression for a bit field in a register assumes that the register exists and does not include the conditions for existence of the register.

### Offset

The term *Offset* is synonymous with *Address*.

### Memory Access Attributes

The Memory Access attribute is defined as <ReadBehavior>/<WriteBehavior> which are defined in the following table.

**Table 13-1      Possible Read and Write Behaviors**

Read (or Write) Behavior	Description
RC	A read clears this register field.
RS	A read sets this register field.
RM	A read modifies the contents of this register field in a manner not described by any of the above.
Wo	You can only write once to this register field.
W1C	A write of 1 clears this register field.
W1S	A write of 1 sets this register field.
W1T	A write of 1 toggles this register field.
W0C	A write of 0 clears this register field.

Read (or Write) Behavior	Description
W0S	A write of 0 sets this register field.
W0T	A write of 0 toggles this register field.
WC	Any write clears this register field.
WS	Any write sets this register field.
WM	A write modifies this register field in a manner not described by any of the above.
no Read Behavior attribute	You cannot read this register. It is Write-Only.
no Write Behavior attribute	You cannot write to this register. It is Read-Only.

**Table 13-2      Memory Access Examples**

Memory Access	Description
R	Read-only register field.
W	Write-only register field.
R/W	Read/write register field.
R/W1C	You can read this register field. Writing 1 clears it.
RC/W1C	Reading this register field clears it. Writing 1 clears it.
R/Wo	You can read this register field. You can only write to it once.

### Special Optional Attributes

Some register fields might use the following optional attributes.

**Table 13-3      Optional Attributes**

Attribute	Description
Volatile	As defined by the IP-XACT specification. If true, indicates in the case of a write followed by read, or in the case of two consecutive reads, there is no guarantee as to what is returned by the read on the second transaction or that this return value is consistent with the write or read of the first transaction. The element implies there is some additional mechanism by which this field can acquire new values other than by reads/writes/resets and other access methods known to IP-XACT. For example, when the IP updates the register field contents.

Attribute	Description
Testable	As defined by the IP-XACT specification. Possible values are unconstrained, untestable, readOnly, writeAsRead, restore. Untestable means that this field is untestable by a simple automated register test. For example, the read-write access of the register is controlled by a pin or another register. readOnly means that you should not write to this register; only read from it. This might apply for a register that modifies the contents of another register.
Reset Mask	As defined by the IP-XACT specification. Indicates that this register field has an unknown reset value. For example, the reset value is set by another register or an input pin; or the register is implemented using RAM.
* Varies	Indicates that the memory access (or reset) attribute (read, write behavior) is not fixed. For example, the read-write access of the register is controlled by a pin or another register. Or when the access depends on some configuration parameter; in this case the post-configuration report in coreConsultant gives the actual access value.

Register definitions in the component memory map.

**Table 13-4 Registers for the dwc\_lpddr5xphy\_top Memory Map**

Register	Offset	Description
DWC_DDRPHYA_DBYTE0_p0 DWC_DDRPHYA_DBYTEj_Pk (for j = 0; j <= DWC_LPDDR5XPHY_NUM_DBYTES-1)(for k = 0; k <= 3) Exists: DWC_LPDDR5XPHY_NUM_DBYTES > j		DesignWare Cores LPDDR5X/5/4X PHY address block
"DFIMRL_pX (for X = 0; X <= 3)" on page 403	(0x10000+(j<<12))+0 x0+(X*0x100000)	DFIMRL_pX: DFI MaxReadLatency
"EnableWriteLinkEcc_pX (for X = 0; X <= 3)" on page 404	(0x10000+(j<<12))+0 x1+(X*0x100000)	EnableWriteLinkEcc_pX: Enable Write Data Link ECC. Applicable only in LPDDR5 Protocol
"DbyteMiscMode" on page 405	(0x10000+(j<<12))+0 x2	DBYTE Module Disable
"DxDfiClkDis_pX (for X = 0; X <= 3)" on page 406	(0x10000+(j<<12))+0 x3+(X*0x100000)	DBYTE DfiClk clock Disable

Register	Offset	Description
"DxPClkDis_pX (for X = 0; X <= 3)" on page 407	(0x10000+(j<<12))+0 x4+(X*0x100000)	DBYTE PClk clock Disable
"LP5DfiDataEnLatency_pX (for X = 0; X <= 3)" on page 408	(0x10000+(j<<12))+0 x8+(X*0x100000)	Selects the depth of the data pipes in the PUB relative to AC pipes
"DfiCtrlRx_fifoRst" on page 409	(0x10000+(j<<12))+0 xb	DfiCtrlRx_fifoRst: Reset RxFifos on sideband transaction for LPDDR54
"PptDqsCntInvTrnTg0_pX (for X = 0; X <= 3)" on page 410	(0x10000+(j<<12))+0 xc+(X*0x100000)	PptDqsCntInvTrnTg0_pX: Programmed by PHY training firmware to support LPDDR4X/5 DRAM drift...
"PptDqsCntInvTrnTg1_pX (for X = 0; X <= 3)" on page 411	(0x10000+(j<<12))+0 xd+(X*0x100000)	PptDqsCntInvTrnTg1_pX: Programmed by PHY training firmware to support LPDDR4X/5 DRAM drift...
"TrackingModeCntrl_pX (for X = 0; X <= 3)" on page 412	(0x10000+(j<<12))+0 xe+(X*0x100000)	Mode controls for fine RxEn timing adjustment during mission-mode operation.
"RxClkT2UIDlyTg0_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)" on page 415	(0x10000+(j<<12))+0 x10+(X*0x100000)+(Y*0x100)	RxClkT2UIDlyTg0_rY_pX: Trained Read DQS to RxClk Delay (Timing Group DEST=0).
"RxClkT2UIDlyTg1_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)" on page 417	(0x10000+(j<<12))+0 x11+(X*0x100000)+(Y*0x100)	RxClkT2UIDlyTg1_rY_pX: Trained Read DQS to RxClk Delay (Timing Group DEST=1).
"RxClkC2UIDlyTg0_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)" on page 419	(0x10000+(j<<12))+0 x12+(X*0x100000)+(Y*0x100)	RxClkC2UIDlyTg0_rY_pX: Trained Read DQS_c to RxClkc Delay (Timing Group DEST=0).
"RxClkC2UIDlyTg1_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)" on page 420	(0x10000+(j<<12))+0 x13+(X*0x100000)+(Y*0x100)	RxClkC2UIDlyTg1_rY_pX: Trained Read DQS_c to RxClkc Delay (Timing Group DEST=1).

Register	Offset	Description
<a href="#">“PptWck2DqoCntInvTrnTg0_pX (for X = 0; X &lt;= 3)” on page 421</a>	(0x10000+ (j<<12))+0 x14+(X*0x 100000)	PptWck2DqoCntInvTrnTg0_pX: Programmed by PHY training firmware to support LPDDR5 DRAM drift...
<a href="#">“PptWck2DqoCntInvTrnTg1_pX (for X = 0; X &lt;= 3)” on page 422</a>	(0x10000+ (j<<12))+0 x15+(X*0x 100000)	PptWck2DqoCntInvTrnTg1_pX: Programmed by PHY training firmware to support LPDDR5 DRAM drift...
<a href="#">“InitSeqControl” on page 423</a>	(0x10000+ (j<<12))+0 x16	AC/DBYTE Pipeline Init Sequence/Pipeline Control register
<a href="#">“TxDqsLeftEyeOffsetTg0_pX (for X = 0; X &lt;= 3)” on page 427</a>	(0x10000+ (j<<12))+0 x19+(X*0x 100000)	TxDqsLeftEyeOffsetTg0_pX: Write DQS Left Eye Offset (Timing Group 0).
<a href="#">“MtestMuxSel” on page 428</a>	(0x10000+ (j<<12))+0 x1a	MtestMuxSel: Digital Observation Pin control
<a href="#">“TxDqsLeftEyeOffsetTg1_pX (for X = 0; X &lt;= 3)” on page 429</a>	(0x10000+ (j<<12))+0 x1b+(X*0x 100000)	TxDqsLeftEyeOffsetTg1_pX: Write DQS Left Eye Offset (Timing Group 1).
<a href="#">“RxEnDlyTg0_pX (for X = 0; X &lt;= 3)” on page 430</a>	(0x10000+ (j<<12))+0 x20+(X*0x 100000)	RxEnDlyTg0_pX: Trained Receive Enable Delay (For Timing Group 0)
<a href="#">“RxEnDlyTg1_pX (for X = 0; X &lt;= 3)” on page 431</a>	(0x10000+ (j<<12))+0 x21+(X*0x 100000)	RxEnDlyTg1_pX: Trained Receive Enable Delay (For Timing Group 1)
<a href="#">“TxDqsRightEyeOffsetTg0_pX (for X = 0; X &lt;= 3)” on page 432</a>	(0x10000+ (j<<12))+0 x22+(X*0x 100000)	TxDqsRightEyeOffsetTg0_pX: Write DQS Right Eye Offset (Timing Group 0).
<a href="#">“TxDqsRightEyeOffsetTg1_pX (for X = 0; X &lt;= 3)” on page 433</a>	(0x10000+ (j<<12))+0 x23+(X*0x 100000)	TxDqsRightEyeOffsetTg1_pX: Write DQS Right Eye Offset (Timing Group 1).
<a href="#">“DqsPreambleControl_pX (for X = 0; X &lt;= 3)” on page 434</a>	(0x10000+ (j<<12))+0 x24+(X*0x 100000)	Control the PHY logic related to the read and write DQS preamble

Register	Offset	Description
"DbyteRxDqsModeCntrl_pX (for X = 0; X <= 3)" on page 436	(0x10000+(j<<12))+0x25+(X*0x100000)	Control for generating RxClk from read DQS edges.
"RxClkCtl1_pX (for X = 0; X <= 3)" on page 438	(0x10000+(j<<12))+0x27+(X*0x100000)	RxClkCtl1_pX: Controls for RxClk timing modes.
"TxDqsDlyTg0_pX (for X = 0; X <= 3)" on page 439	(0x10000+(j<<12))+0x28+(X*0x100000)	TxDqsDlyTg0_pX: Write DQS Delay (Timing Group DEST=0).
"TxDqsDlyTg1_pX (for X = 0; X <= 3)" on page 440	(0x10000+(j<<12))+0x29+(X*0x100000)	TxDqsDlyTg1_pX: Write DQS Delay (Timing Group DEST=1).
"TxWckDlyTg0_pX (for X = 0; X <= 3)" on page 441	(0x10000+(j<<12))+0x2a+(X*0x100000)	TxWckDlyTg0_pX: Write WCK Delay (Timing Group DEST=0).
"TxWckDlyTg1_pX (for X = 0; X <= 3)" on page 442	(0x10000+(j<<12))+0x2b+(X*0x100000)	TxWckDlyTg1_pX: Write WCK Delay (Timing Group DEST=1).
"WrLevBits" on page 443	(0x10000+(j<<12))+0x2e	Write level feedback DQ observability select.
"NeverGateDBDlyCalValClk" on page 444	(0x10000+(j<<12))+0x2f	NeverGateDBDlyCalValClk: Reserved for PHY training firmware use.
"RxClkCtl" on page 445	(0x10000+(j<<12))+0x31	RxClkCtl: Controls for RxClk timing modes.
"RxModeCtlRxReplica_pX (for X = 0; X <= 3)" on page 446	(0x10000+(j<<12))+0x39+(X*0x100000)	RxModeCtlRxReplica_pX: Deprecated
"RxGainCurrAdjRxReplica_pX (for X = 0; X <= 3)" on page 447	(0x10000+(j<<12))+0x3e+(X*0x100000)	RxGainCurrAdjRxReplica_pX: Deprecated

Register	Offset	Description
"TtcfControl" on page 448	(0x10000+(j<<12))+0x3f	TTCF interface control
"LcdlCalControl" on page 449	(0x10000+(j<<12))+0x47	DLL Lock State machine control register
"NeverGateTrainCntrClk" on page 450	(0x10000+(j<<12))+0x49	NeverGateTrainCntrClk: Reserved for PHY training firmware use.
"DBYTEParityInvert" on page 451	(0x10000+(j<<12))+0x4d	DBYTEParityInvert: Invert APB Parity for register slave DBYTE
"RdfPtrChkControl" on page 452	(0x10000+(j<<12))+0x5d	Controls for the checkers for the read-data-FIFOs read and write pointers.
"DxRxStandbyEn_pX (for X = 0; X <= 3)" on page 454	(0x10000+(j<<12))+0x5f+(X*0x10000)	DxRxStandbyEn_pX: Per DBYTE RxStandby Control.
"TxDqLeftEyeOffsetTg0_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)" on page 455	(0x10000+(j<<12))+0x60+(X*0x10000)+(Y*0x100)	TxDqLeftEyeOffsetTg0_rY_pX: Write DQ Left Eye Offset (Timing Group 0).
"TxDqLeftEyeOffsetTg1_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)" on page 457	(0x10000+(j<<12))+0x61+(X*0x10000)+(Y*0x100)	TxDqLeftEyeOffsetTg1_rY_pX: Write DQ Left Eye Offset (Timing Group 1).
"TxDqRightEyeOffsetTg0_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)" on page 459	(0x10000+(j<<12))+0x63+(X*0x10000)+(Y*0x100)	TxDqRightEyeOffsetTg0_rY_pX: Write DQ Right Eye Offset (Timing Group 0).
"TxDqRightEyeOffsetTg1_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)" on page 461	(0x10000+(j<<12))+0x64+(X*0x10000)+(Y*0x100)	TxDqRightEyeOffsetTg1_rY_pX: Write DQ Right Eye Offset (Timing Group 1).
"TrainingCntrSnap_rX (for X = 0; X <= 8)" on page 463	(0x10000+(j<<12))+0x65+(X*0x100)	TrainingCntrSnap_rX: Capture current contents of certain training registers.

Register	Offset	Description
"DtsmErrCountSnap_iX (for X = 0; X <= 8)" on page 464	(0x10000+(j<<12))+0x66+(X*0x100)	DtsmErrCountSnap_iX: Capture current contents of certain training registers.
"DtsmGoodCountSnap_iX (for X = 0; X <= 8)" on page 465	(0x10000+(j<<12))+0x67+(X*0x100)	DtsmGoodCountSnap_iX: Capture current contents of certain training registers.
"RxClkTLeftEyeOffsetTg0_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)" on page 466	(0x10000+(j<<12))+0x68+(X*0x100000)+(Y*0x100)	RxClkTLeftEyeOffsetTg0_rY_pX: RxClkT Left Eye Offset (Timing Group 0).
"RxClkTLeftEyeOffsetTg1_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)" on page 468	(0x10000+(j<<12))+0x69+(X*0x100000)+(Y*0x100)	RxClkTLeftEyeOffsetTg1_rY_pX: RxClkT Left Eye Offset (Timing Group 1).
"RxClkTRightEyeOffsetTg0_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)" on page 470	(0x10000+(j<<12))+0x6a+(X*0x100000)+(Y*0x100)	RxClkTRightEyeOffsetTg0_rY_pX: RxClkT Right Eye Offset (Timing Group 0).
"RxClkTRightEyeOffsetTg1_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)" on page 472	(0x10000+(j<<12))+0x6b+(X*0x100000)+(Y*0x100)	RxClkTRightEyeOffsetTg1_rY_pX: RxClkT Right Eye Offset (Timing Group 1).
"RxClkCLeftEyeOffsetTg0_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)" on page 474	(0x10000+(j<<12))+0x6c+(X*0x100000)+(Y*0x100)	RxClkCLeftEyeOffsetTg0_rY_pX: RxClkC Left Eye Offset (Timing Group 0).
"RxClkCLeftEyeOffsetTg1_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)" on page 476	(0x10000+(j<<12))+0x6d+(X*0x100000)+(Y*0x100)	RxClkCLeftEyeOffsetTg1_rY_pX: RxClkC Left Eye Offset (Timing Group 1).
"RxClkCRightEyeOffsetTg0_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)" on page 478	(0x10000+(j<<12))+0x6e+(X*0x100000)+(Y*0x100)	RxClkCRightEyeOffsetTg0_rY_pX: RxClkC Right Eye Offset (Timing Group 0).

Register	Offset	Description
"RxClkCRightEyeOffsetTg1_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)" on page 480	(0x10000+ (j<<12))+0 x6f+(X*0x 100000)+(Y*0x100)	RxClkCRightEyeOffsetTg1_rY_pX: RxClkC Right Eye Offset (Timing Group 1).
"RxFifoVisibility" on page 482	(0x10000+ (j<<12))+0 x72	RX FIFO visibility
"RxFifoContents_rX (for X = 0; X <= 8)" on page 484	(0x10000+ (j<<12))+0 x73+(X*0x 100)	RxFifoContents_rX: Per Lane RX FIFO contents
"TrainingCntr_rX (for X = 0; X <= 8)" on page 485	(0x10000+ (j<<12))+0 x74+(X*0x 100)	TrainingCntr_rX: Reserved for PHY training firmware use.
"WckDiffCtl" on page 486	(0x10000+ (j<<12))+0 x75	CSR control for WCK DIFF slices
"PptRxClkInfo_rX (for X = 0; X <= 8)" on page 487	(0x10000+ (j<<12))+0 x77+(X*0x 100)	PptRxClkInfo_rX: Reserved for PHY training firmware use.
"RxDigStrbDlyTg0_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)" on page 488	(0x10000+ (j<<12))+0 x78+(X*0x 100000)+(Y*0x100)	RxDigStrbDlyTg0_rY_pX: Rx Digital Strobe Coarse Delay (Timing Group 0).
"RxDigStrbDlyTg1_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)" on page 490	(0x10000+ (j<<12))+0 x79+(X*0x 100000)+(Y*0x100)	RxDigStrbDlyTg1_rY_pX: Rx Digital Strobe Coarse Delay (Timing Group 1).
"TxDqDlyTg0_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)" on page 492	(0x10000+ (j<<12))+0 x7a+(X*0x 100000)+(Y*0x100)	TxDqDlyTg0_rY_pX: Write DQ Delay (Timing Group 0).
"TxDqDlyTg1_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)" on page 494	(0x10000+ (j<<12))+0 x7b+(X*0x 100000)+(Y*0x100)	TxDqDlyTg1_rY_pX: Write DQ Delay (Timing Group 1).

Register	Offset	Description
"SingleEndedMode_pX (DBYTE) (for X = 0; X <= 3)" on page 496	(0x10000+(j<<12))+0 x7c+(X*0x10000)	Control for Single Ended LP4X/5 Mode
"ScratchPadDBYTE" on page 497	(0x10000+(j<<12))+0 x7d	ScratchPadDBYTE: ScratchPad for DBYTE
"Dq0LnSel" on page 498	(0x10000+(j<<12))+0 x80	Dq0LnSel: Maps Phy DQ lane to memory DQ0
"Dq1LnSel" on page 499	(0x10000+(j<<12))+0 x81	Dq1LnSel: Maps Phy DQ lane to memory DQ1
"Dq2LnSel" on page 500	(0x10000+(j<<12))+0 x82	Dq2LnSel: Maps Phy DQ lane to memory DQ2
"Dq3LnSel" on page 501	(0x10000+(j<<12))+0 x83	Dq3LnSel: Maps Phy DQ lane to memory DQ3
"Dq4LnSel" on page 502	(0x10000+(j<<12))+0 x84	Dq4LnSel: Maps Phy DQ lane to memory DQ4
"Dq5LnSel" on page 503	(0x10000+(j<<12))+0 x85	Dq5LnSel: Maps Phy DQ lane to memory DQ5
"Dq6LnSel" on page 504	(0x10000+(j<<12))+0 x86	Dq6LnSel: Maps Phy DQ lane to memory DQ6
"Dq7LnSel" on page 505	(0x10000+(j<<12))+0 x87	Dq7LnSel: Maps Phy DQ lane to memory DQ7
"Dq8LnSel" on page 506	(0x10000+(j<<12))+0 x88	Dq8LnSel: Maps Phy DQ lane to memory DQ8
"AsyncDbyteTxMode" on page 507	(0x10000+(j<<12))+0 x89	AsyncDbyteTxMode: Reserved for PHY training firmware use.
"AsyncDbyteRxMode" on page 508	(0x10000+(j<<12))+0 x8a	AsyncDbyteRxMode: Reserved for PHY training firmware use.

Register	Offset	Description
" <a href="#">AsyncDbyteTxEn</a> " on page 509	(0x10000+(j<<12))+0x8b	AsyncDbyteTxEn: Reserved for PHY training firmware use.
" <a href="#">AsyncDbyteTxData</a> " on page 510	(0x10000+(j<<12))+0x8c	AsyncDbyteTxData: Reserved for PHY training firmware use.
" <a href="#">AsyncDbyteRxData</a> " on page 511	(0x10000+(j<<12))+0x8d	AsyncDbyteRxData: Reserved for PHY training firmware use.
" <a href="#">SelDbCurDlyTmngInfo</a> " on page 512	(0x10000+(j<<12))+0x8e	SelDbCurDlyTmngInfo: Reserved for PHY training firmware use and for debugging
" <a href="#">DxLoopBackEn</a> " on page 513	(0x10000+(j<<12))+0x92	DxLoopBackEn: Per DBYTE Loopback Enable Configuration
" <a href="#">DxDigStrobeGenSel</a> " on page 514	(0x10000+(j<<12))+0x94	DxDigStrobeGenSel: Selects the source for the generation of the RxDigStrobe
" <a href="#">DxDigStrobePat</a> " on page 515	(0x10000+(j<<12))+0x95	DxDigStrobePat: Selects the pattern for RxDigStrobe
" <a href="#">DxRxStrobeEnPatWck</a> " on page 516	(0x10000+(j<<12))+0x96	DxRxStrobeEnPatWck: Selects the pattern for RxStrobeEn for WCK DIFF Slice.
" <a href="#">DxDtEn</a> " on page 517	(0x10000+(j<<12))+0x97	DBYTE OdtEn control
" <a href="#">DxDsampleDIFF</a> " on page 518	(0x10000+(j<<12))+0x98	Address Loopback sample values for Diff slice
" <a href="#">RxClkTrkErr00</a> " on page 520	(0x10000+(j<<12))+0x99	For RxClkDly tracking of tPHY_tDQS2DQ; overflow, underflow for Tg0,Tg1
" <a href="#">PptInfoSel</a> " on page 524	(0x10000+(j<<12))+0x9e	PptInfoSel: Reserved for PHY training firmware use.
" <a href="#">Wck2DqoPptInfo</a> " on page 525	(0x10000+(j<<12))+0x9f	Wck2DqoPptInfo: Reserved for PHY training firmware use.

Register	Offset	Description
"Dqs2DqPptInfo" on page 526	(0x10000+ (j<<12))+0 xa0	Dqs2DqPptInfo: Reserved for PHY training firmware use.
"PptRxDqsTrackInfo" on page 527	(0x10000+ (j<<12))+0 xa1	PptRxDqsTrackInfo: Reserved for PHY training firmware use.
"PptRxEnEvnt" on page 528	(0x10000+ (j<<12))+0 xa2	PptRxEnEvnt: This register is dynamically written by PHY Initialization Engine during frequency...
"PptCtlStatic" on page 529	(0x10000+ (j<<12))+0 xa3	Controls for the PPT of tDQS2DQ
"PptCtlDyn" on page 531	(0x10000+ (j<<12))+0 xa4	PptCtlDyn: This register is dynamically written by PHY Initialization Engine during frequency changes...
"RxTrainPattern8BitMode_pX (for X = 0; X <= 3)" on page 532	(0x10000+ (j<<12))+0 xa5+(X*0x 100000)	RxTrainPattern8BitMode_pX: Reserved for PHY training firmware use.
"RxTrainPatternEnable" on page 533	(0x10000+ (j<<12))+0 xa6	RxTrainPatternEnable: Reserved for PHY training firmware use.
"TrainingParam" on page 534	(0x10000+ (j<<12))+0 xa7	TrainingParam: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support LPDDR4X DRAM drift...
"DtshmByteCtrl0" on page 535	(0x10000+ (j<<12))+0 xb0	DtshmByteCtrl0: Reserved for PHY training firmware use.
"DtshmByteCtrl1" on page 536	(0x10000+ (j<<12))+0 xb1	DtshmByteCtrl1: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() for protocols that support DRAM...
"TxChkDataSelects" on page 537	(0x10000+ (j<<12))+0 xb2	TxChkDataSelects: Reserved for PHY training firmware use.
"DtshmGateInc" on page 538	(0x10000+ (j<<12))+0 xb3	DtshmGateInc: Reserved for PHY training firmware use.
"DtshmGateDec" on page 539	(0x10000+ (j<<12))+0 xb4	DtshmGateDec: Reserved for PHY training firmware use.

Register	Offset	Description
"DtsmLaneCtrl0_iX (for X = 0; X <= 8)" on page 540	(0x10000+(j<<12))+0 xb5+(X*0x100)	DtsmLaneCtrl0_iX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support LPDDR4X DRAM drift...
"DtsmCmpCount_iX (for X = 0; X <= 8)" on page 541	(0x10000+(j<<12))+0 xb6+(X*0x100)	DtsmCmpCount_iX: Reserved for PHY training firmware use.
"DtsmErrCount_iX (for X = 0; X <= 8)" on page 542	(0x10000+(j<<12))+0 xb7+(X*0x100)	DtsmErrCount_iX: Reserved for PHY training firmware use.
"DtsmGoodCount_iX (for X = 0; X <= 8)" on page 543	(0x10000+(j<<12))+0 xb8+(X*0x100)	DtsmGoodCount_iX: Reserved for PHY training firmware use.
"DtsmGoodBar" on page 544	(0x10000+(j<<12))+0 xb9	DtsmGoodBar: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support LPDDR4X DRAM drift...
"DtsmErrBar" on page 545	(0x10000+(j<<12))+0 xba	DtsmErrBar: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support LPDDR4X DRAM drift...
"DtsmCountClears" on page 546	(0x10000+(j<<12))+0 xbb	DtsmCountClears: Reserved for PHY training firmware use.
"DtsmGoodThldXingInd" on page 547	(0x10000+(j<<12))+0 xbc	DtsmGoodThldXingInd: Reserved for PHY training firmware use.
"DtsmErrThldXingInd" on page 548	(0x10000+(j<<12))+0 xbd	DtsmErrThldXingInd: Reserved for PHY training firmware use.
"TrainingIncDecDtsmEn_rX (for X = 0; X <= 8)" on page 549	(0x10000+(j<<12))+0 xbe+(X*0x100)	TrainingIncDecDtsmEn_rX: Reserved for PHY training firmware use.
"DqRxVrefDac_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)" on page 550	(0x10000+(j<<12))+0 xc8+(X*0x10000)+(Y*0x100)	DqRxVrefDac_rY_pX: Rx VREF control for a TXRXDQ

Register	Offset	Description
"RxReplicaLcdlPh1UI2UI" on page 551	(0x10000+(j<<12))+0 xd5	RxReplicaLcdlPh1UI2UI: For RxClkDly tracking of tPHY_tDQS2DQ; RxReplica LCDL
"RxReplicaRatioNow" on page 552	(0x10000+(j<<12))+0 xd6	RxReplicaRatioNow: For RxClkDly tracking of tPHY_tDQS2DQ; term used in correction.
"RxReplicaRxClockDlyCorrection" on page 553	(0x10000+(j<<12))+0 xd7	For RxClkDly tracking of tPHY_tDQS2DQ; correction value.
"RxReplicaStatus00" on page 555	(0x10000+(j<<12))+0 xd8	For RxClkDly tracking of tPHY_tDQS2DQ; valid signal for csr read values.
"RxReplicaUICalWait" on page 556	(0x10000+(j<<12))+0 xd9	RxReplicaUICalWait: RxReplica UI calibration wait time
"RxReplicaInterval" on page 557	(0x10000+(j<<12))+0 xda	RxReplicaInterval: RxReplica calibration interval time
"RxReplicaDontGateRCTMath" on page 558	(0x10000+(j<<12))+0 xdb	RxReplicaDontGateRCTMath: Keep this zero. If 1, disables the power saving logic for RCT Math
"DbyteCntrl" on page 559	(0x10000+(j<<12))+0 xde	Misc Control CSR for DBYTE
"DbCurrentDlyTimingInfoTgX (for X = 0; X <= 1)" on page 560	(0x10000+(j<<12))+0 xe4+(X*0x1)	DbCurrentDlyTimingInfoTgX: Reserved for PHY training firmware use and for debugging
"DxRxPowerDown" on page 561	(0x10000+(j<<12))+0 xfa	Dq/Dqs receiver control
"RxDigStrbEn_pX (for X = 0; X <= 3)" on page 562	(0x10000+(j<<12))+0 xfb+(X*0x100000)	Enable Digital Strobe Read Mode.
"DxPipeEn_pX (for X = 0; X <= 3)" on page 564	(0x10000+(j<<12))+0 xfc+(X*0x100000)	DxPipeEn_pX: Deprecated

Register	Offset	Description
" <a href="#">LcdlCalCtrl</a> " on page 565	(0x10000+ (j<<12))+0 xfe	LcdlCalCtrl: Reserved for PHY test firmware use.
" <a href="#">PclkDCDCtrl_pX (DBYTE) (for X = 0; X &lt;= 3)" on page 566</a>	(0x10000+ (j<<12))+0 x100+(X*0 x100000)	PclkDCDCtrl_pX: Controls the DCD Comparator for DCA calibration
" <a href="#">PPTTrainSetup2_pX (for X = 0; X &lt;= 3)" on page 567</a>	(0x10000+ (j<<12))+0 x102+(X*0 x100000)	PPTTrainSetup2_pX: Programmed by PHY training firmware to support LPDDR4X DRAM drift...
" <a href="#">ForceInternalUpdate</a> " on page 568	(0x10000+ (j<<12))+0 x103	ForceInternalUpdate: This Register used by Training Firmware to force an internal PHY Update...
" <a href="#">ForceRxDataFifoUpd</a> " on page 569	(0x10000+ (j<<12))+0 x104	ForceRxDataFifoUpd: This register used to force update event in RxDataFifo
" <a href="#">DMIPinPresent_pX (for X = 0; X &lt;= 3)" on page 570</a>	(0x10000+ (j<<12))+0 x108+(X*0 x100000)	This Register is used to enable the Read-DBI function in each DBYTE
" <a href="#">InhibitTxRdPtrInit_pX (for X = 0; X &lt;= 3)" on page 571</a>	(0x10000+ (j<<12))+0 x10b+(X*0 x100000)	InhibitTxRdPtrInit_pX: For use by Training FW
" <a href="#">AllowInhibitTxRdPtrInit</a> " on page 572	(0x10000+ (j<<12))+0 x10c	AllowInhibitTxRdPtrInit: For use by Training FW
" <a href="#">RdfPtrChkStatusWptrT</a> " on page 573	(0x10000+ (j<<12))+0 x15d	RdfPtrChkStatusWptrT: Error status for RDF WptrT checker
" <a href="#">RDqRDqsCntrl_pX (for X = 0; X &lt;= 3)" on page 574</a>	(0x10000+ (j<<12))+0 x15f+(X*0 x100000)	Dq/Dqs receiver control
" <a href="#">RxReplicaLcdlCalCtrl</a> " on page 576	(0x10000+ (j<<12))+0 x189	RxReplicaLcdlCalCtrl: Reserved for PHY test firmware use.
" <a href="#">RxReplicaRangeVal_pX (for X = 0; X &lt;= 3)" on page 577</a>	(0x10000+ (j<<12))+0 x209+(X*0 x100000)	RxReplicaRangeVal_pX: Controls for RxClk timing modes.

Register	Offset	Description
"RxReplicaCtl04_pX (for X = 0; X <= 3)" on page 578	(0x10000+(j<<12))+0 x20f+(X*0 x100000)	For RxClkDly tracking of tPHY_tDQS2DQ; control for RxReplica
"RdfPtrChkStatusWptrC" on page 580	(0x10000+(j<<12))+0 x25d	RdfPtrChkStatusWptrC: Error status for RDF WptrT checker
"RxReplicaPathPhase0_pX (for X = 0; X <= 3)" on page 581	(0x10000+(j<<12))+0 x2a0+(X*0 x100000)	RxReplicaPathPhase0_pX: For RxClkDly tracking of tPHY_tDQS2DQ; RxReplica path delay
"RxReplicaPathPhase1_pX (for X = 0; X <= 3)" on page 582	(0x10000+(j<<12))+0 x2a1+(X*0 x100000)	RxReplicaPathPhase1_pX: For RxClkDly tracking of tPHY_tDQS2DQ; RxReplica path delay
"RxReplicaPathPhase2_pX (for X = 0; X <= 3)" on page 583	(0x10000+(j<<12))+0 x2a2+(X*0 x100000)	RxReplicaPathPhase2_pX: For RxClkDly tracking of tPHY_tDQS2DQ; RxReplica path delay
"RxReplicaPathPhase3_pX (for X = 0; X <= 3)" on page 584	(0x10000+(j<<12))+0 x2a3+(X*0 x100000)	RxReplicaPathPhase3_pX: For RxClkDly tracking of tPHY_tDQS2DQ; RxReplica path delay
"RxReplicaPathPhase4_pX (for X = 0; X <= 3)" on page 585	(0x10000+(j<<12))+0 x2a4+(X*0 x100000)	RxReplicaPathPhase4_pX: For RxClkDly tracking of tPHY_tDQS2DQ; RxReplica path delay
"RxReplicaCtl00" on page 586	(0x10000+(j<<12))+0 x2ac	For RxClkDly tracking of tPHY_tDQS2DQ; for reading status
"RxReplicaCtl01_pX (for X = 0; X <= 3)" on page 587	(0x10000+(j<<12))+0 x2ad+(X*0 x100000)	For RxClkDly tracking of tPHY_tDQS2DQ; control for RxReplica
"RxReplicaCtl02_pX (for X = 0; X <= 3)" on page 588	(0x10000+(j<<12))+0 x2ae+(X*0 x100000)	For RxClkDly tracking of tPHY_tDQS2DQ; correction limit.
"RxReplicaCtl03_pX (for X = 0; X <= 3)" on page 589	(0x10000+(j<<12))+0 x2af+(X*0 x100000)	For RxClkDly tracking of tPHY_tDQS2DQ; Baseline value.

Register	Offset	Description
"RdfPtrChkStatusRptr" on page 590	(0x10000+(j<<12))+0x35d	RdfPtrChkStatusRptr: Error status for RDF WptrT checker
"PclkDCACalCtrl0DB" on page 591	(0x10000+(j<<12))+0x800	Controls DCA sample polarity
"PclkDCADynCtrl" on page 592	(0x10000+(j<<12))+0x802	Dynamic bits for DCA control
"PclkDCASStaticCtrl0DB_pX (for X = 0; X <= 3)" on page 593	(0x10000+(j<<12))+0x803+(X*0x100000)	Static bits for DCA control in DBYTE blocks
"PclkDCASampCntDB" on page 595	(0x10000+(j<<12))+0x804	Sample counts for various stages of DCA calibration
"PclkDCAHysMaskDB" on page 596	(0x10000+(j<<12))+0x805	PclkDCAHysMaskDB: Used by DCA quick search algorithm
"PclkDCACalFineBoundDB" on page 597	(0x10000+(j<<12))+0x806	Sets limit values of DCA Fine searches
"PclkDCANextFineOnCoarseDB" on page 598	(0x10000+(j<<12))+0x807	Sets the DCAFine value to use when Coarse changes
"PclkDCAFullSearchIVACDB" on page 599	(0x10000+(j<<12))+0x808	Initial DCA Fine value to use for full searches
"PclkDCASampDelayLCDLDB_pX (for X = 0; X <= 3)" on page 600	(0x10000+(j<<12))+0x80b+(X*0x100000)	PclkDCASampDelayLCDLDB_pX: Additional delay for SampVld generation in LCDL mode
"LcdITstCtrl" on page 601	(0x10000+(j<<12))+0x884	LcdITstCtrl: Reserved for PHY test firmware use.
"DtssErrCountNZ" on page 602	(0x10000+(j<<12))+0x9bf	DtssErrCountNZ: Reserved for PHY training firmware use.

Register	Offset	Description
" <a href="#">PclkDCASStaticCtrl1DB_pX (for X = 0; X &lt;= 3)" on page 603</a>	(0x10000+ (j<<12))+0 xc03+(X*0 x100000)	Static bits for DCA control in DBYTE blocks
" <a href="#">PclkDCACoarseBoundDB" on page 604</a>	(0x10000+ (j<<12))+0 xc1d	Sets limit values of DCA Coarse searches
" <a href="#">PclkDCAMiscCtrlDB" on page 605</a>	(0x10000+ (j<<12))+0 xc1f	PclkDCAMiscCtrlDB: Misc DCA control bits
" <a href="#">TrainingResultsSnap" on page 606</a>	(0x10000+ (j<<12))+0 xd27	TrainingResultsSnap: Enable for capture current contents of certain training registers.
" <a href="#">PpgcChkCtrl" on page 607</a>	(0x10000+ (j<<12))+0 xe00	PpgcChkCtrl: Reserved for PHY training firmware use.
" <a href="#">PpgcChkDbiCtrl" on page 608</a>	(0x10000+ (j<<12))+0 xe01	PpgcChkDbiCtrl: Reserved for PHY training firmware use.
" <a href="#">PpgcChkDbiConfig" on page 609</a>	(0x10000+ (j<<12))+0 xe02	PpgcChkDbiConfig: Reserved for PHY training firmware use.
" <a href="#">PpgcChkLaneMuxSel0" on page 610</a>	(0x10000+ (j<<12))+0 xe03	PpgcChkLaneMuxSel0: Reserved for PHY training firmware use.
" <a href="#">PpgcChkLaneMuxSel1" on page 611</a>	(0x10000+ (j<<12))+0 xe04	PpgcChkLaneMuxSel1: Reserved for PHY training firmware use.
" <a href="#">PpgcChkTxManipulationCtrl0" on page 612</a>	(0x10000+ (j<<12))+0 xe08	PpgcChkTxManipulationCtrl0: Reserved for PHY training firmware use.
" <a href="#">PpgcChkTxManipulationCtrl1" on page 613</a>	(0x10000+ (j<<12))+0 xe09	PpgcChkTxManipulationCtrl1: Reserved for PHY training firmware use.
" <a href="#">PpgcChkTxManipulationCtrl2" on page 614</a>	(0x10000+ (j<<12))+0 xe0a	PpgcChkTxManipulationCtrl2: Reserved for PHY training firmware use.
" <a href="#">PpgcChkTxManipluationPrbs9" on page 615</a>	(0x10000+ (j<<12))+0 xe0b	PpgcChkTxManipluationPrbs9: Reserved for PHY training firmware use.

Register	Offset	Description
" <a href="#">PpgcChkRxManipulationCtrl0</a> " on page <a href="#">616</a>	(0x10000+ (j<<12))+0 xe0c	PpgcChkRxManipulationCtrl0: Reserved for PHY training firmware use.
" <a href="#">PpgcChkRxManipulationCtrl1</a> " on page <a href="#">617</a>	(0x10000+ (j<<12))+0 xe0d	PpgcChkRxManipulationCtrl1: Reserved for PHY training firmware use.
" <a href="#">PpgcChkRxManipulationCtrl2</a> " on page <a href="#">618</a>	(0x10000+ (j<<12))+0 xe0e	PpgcChkRxManipulationCtrl2: Reserved for PHY training firmware use.
" <a href="#">PpgcChkRxManipulationPrbs9</a> " on page <a href="#">619</a>	(0x10000+ (j<<12))+0 xe0f	PpgcChkRxManipulationPrbs9: Reserved for PHY training firmware use.
" <a href="#">PclkDCAClkGaterEnDB</a> " on page <a href="#">620</a>	(0x10000+ (j<<12))+0 xe1f	PclkDCAClkGaterEnDB: 1 = Clk to DCA FSMs is enabled
" <a href="#">Prbs0ChkModeSel</a> " on page <a href="#">621</a>	(0x10000+ (j<<12))+0 xe20	Prbs0ChkModeSel: Reserved for PHY training firmware use.
" <a href="#">Prbs0ChkUiMuxSel</a> " on page <a href="#">622</a>	(0x10000+ (j<<12))+0 xe21	Prbs0ChkUiMuxSel: Reserved for PHY training firmware use.
" <a href="#">Prbs0ChkTapDly0</a> " on page <a href="#">623</a>	(0x10000+ (j<<12))+0 xe22	Prbs0ChkTapDly0: Reserved for PHY training firmware use.
" <a href="#">Prbs0ChkTapDly1</a> " on page <a href="#">624</a>	(0x10000+ (j<<12))+0 xe23	Prbs0ChkTapDly1: Reserved for PHY training firmware use.
" <a href="#">Prbs0ChkTapDly2</a> " on page <a href="#">625</a>	(0x10000+ (j<<12))+0 xe24	Prbs0ChkTapDly2: Reserved for PHY training firmware use.
" <a href="#">Prbs0ChkTapDly3</a> " on page <a href="#">626</a>	(0x10000+ (j<<12))+0 xe25	Prbs0ChkTapDly3: Reserved for PHY training firmware use.
" <a href="#">Prbs0ChkTapDly4</a> " on page <a href="#">627</a>	(0x10000+ (j<<12))+0 xe26	Prbs0ChkTapDly4: Reserved for PHY training firmware use.
" <a href="#">Prbs0ChkTapDly5</a> " on page <a href="#">628</a>	(0x10000+ (j<<12))+0 xe27	Prbs0ChkTapDly5: Reserved for PHY training firmware use.

Register	Offset	Description
"Prbs0ChkTapDly6" on page 629	(0x10000+(j<<12))+0 xe28	Prbs0ChkTapDly6: Reserved for PHY training firmware use.
"Prbs0ChkTapDly7" on page 630	(0x10000+(j<<12))+0 xe29	Prbs0ChkTapDly7: Reserved for PHY training firmware use.
"Prbs0ChkStateLo" on page 631	(0x10000+(j<<12))+0 xe2a	Prbs0ChkStateLo: Reserved for PHY training firmware use.
"Prbs0ChkStateHi" on page 632	(0x10000+(j<<12))+0 xe2b	Prbs0ChkStateHi: Reserved for PHY training firmware use.
"Prbs1ChkModeSel" on page 633	(0x10000+(j<<12))+0 xe30	Prbs1ChkModeSel: Reserved for PHY training firmware use.
"Prbs1ChkUiMuxSel" on page 634	(0x10000+(j<<12))+0 xe31	Prbs1ChkUiMuxSel: Reserved for PHY training firmware use.
"Prbs1ChkTapDly0" on page 635	(0x10000+(j<<12))+0 xe32	Prbs1ChkTapDly0: Reserved for PHY training firmware use.
"Prbs1ChkTapDly1" on page 636	(0x10000+(j<<12))+0 xe33	Prbs1ChkTapDly1: Reserved for PHY training firmware use.
"Prbs1ChkTapDly2" on page 637	(0x10000+(j<<12))+0 xe34	Prbs1ChkTapDly2: Reserved for PHY training firmware use.
"Prbs1ChkTapDly3" on page 638	(0x10000+(j<<12))+0 xe35	Prbs1ChkTapDly3: Reserved for PHY training firmware use.
"Prbs1ChkTapDly4" on page 639	(0x10000+(j<<12))+0 xe36	Prbs1ChkTapDly4: Reserved for PHY training firmware use.
"Prbs1ChkTapDly5" on page 640	(0x10000+(j<<12))+0 xe37	Prbs1ChkTapDly5: Reserved for PHY training firmware use.
"Prbs1ChkTapDly6" on page 641	(0x10000+(j<<12))+0 xe38	Prbs1ChkTapDly6: Reserved for PHY training firmware use.

Register	Offset	Description
"Prbs1ChkTapDly7" on page 642	(0x10000+ (j<<12))+0 xe39	Prbs1ChkTapDly7: Reserved for PHY training firmware use.
"Prbs1ChkStateLo" on page 643	(0x10000+ (j<<12))+0 xe3a	Prbs1ChkStateLo: Reserved for PHY training firmware use.
"Prbs1ChkStateHi" on page 644	(0x10000+ (j<<12))+0 xe3b	Prbs1ChkStateHi: Reserved for PHY training firmware use.
"Prbs2ChkModeSel" on page 645	(0x10000+ (j<<12))+0 xe40	Prbs2ChkModeSel: Reserved for PHY training firmware use.
"Prbs2ChkUiMuxSel" on page 646	(0x10000+ (j<<12))+0 xe41	Prbs2ChkUiMuxSel: Reserved for PHY training firmware use.
"Prbs2ChkTapDly0" on page 647	(0x10000+ (j<<12))+0 xe42	Prbs2ChkTapDly0: Reserved for PHY training firmware use.
"Prbs2ChkTapDly1" on page 648	(0x10000+ (j<<12))+0 xe43	Prbs2ChkTapDly1: Reserved for PHY training firmware use.
"Prbs2ChkTapDly2" on page 649	(0x10000+ (j<<12))+0 xe44	Prbs2ChkTapDly2: Reserved for PHY training firmware use.
"Prbs2ChkTapDly3" on page 650	(0x10000+ (j<<12))+0 xe45	Prbs2ChkTapDly3: Reserved for PHY training firmware use.
"Prbs2ChkTapDly4" on page 651	(0x10000+ (j<<12))+0 xe46	Prbs2ChkTapDly4: Reserved for PHY training firmware use.
"Prbs2ChkTapDly5" on page 652	(0x10000+ (j<<12))+0 xe47	Prbs2ChkTapDly5: Reserved for PHY training firmware use.
"Prbs2ChkTapDly6" on page 653	(0x10000+ (j<<12))+0 xe48	Prbs2ChkTapDly6: Reserved for PHY training firmware use.
"Prbs2ChkTapDly7" on page 654	(0x10000+ (j<<12))+0 xe49	Prbs2ChkTapDly7: Reserved for PHY training firmware use.

Register	Offset	Description
"Prbs2ChkStateLo" on page 655	(0x10000+ (j<<12))+0 xe4a	Prbs2ChkStateLo: Reserved for PHY training firmware use.
"Prbs2ChkStateHi" on page 656	(0x10000+ (j<<12))+0 xe4b	Prbs2ChkStateHi: Reserved for PHY training firmware use.
"PpgcChkFltCfg0" on page 657	(0x10000+ (j<<12))+0 xea0	PpgcChkFltCfg0: Reserved for PHY training firmware use.
"PpgcChkFltCfg1" on page 658	(0x10000+ (j<<12))+0 xea1	PpgcChkFltCfg1: Reserved for PHY training firmware use.
"PpgcChkFltCfg2" on page 659	(0x10000+ (j<<12))+0 xea2	PpgcChkFltCfg2: Reserved for PHY training firmware use.
"PpgcChkMskPat0" on page 660	(0x10000+ (j<<12))+0 xea4	PpgcChkMskPat0: Reserved for PHY training firmware use.
"PpgcChkMskPat1" on page 661	(0x10000+ (j<<12))+0 xea5	PpgcChkMskPat1: Reserved for PHY training firmware use.
"PpgcChkMskPat2" on page 662	(0x10000+ (j<<12))+0 xea6	PpgcChkMskPat2: Reserved for PHY training firmware use.
"PpgcChkMskPat3" on page 663	(0x10000+ (j<<12))+0 xea7	PpgcChkMskPat3: Reserved for PHY training firmware use.
DWC_DDRPHYA_MASTER0_p0 DesignWare Cores LPDDR5X/5/4X PHY address block DWC_DDRPHYA_MASTERj_Pk (for j == 0)(for k = 0; k <= 3) Exists: Always		
"DfiFreqRatio_pX (for X = 0; X <= 3)" on page 665	0x20000+ 0x0+(X*0x 100000)	DfiFreqRatio_pX: DFI Frequency Ratio
"PubDxPowerDownControl" on page 666	0x20000+ 0x1	PubDxPowerDownControl: Bits to turn off the PUB_DX
"PclkPtrInitVal_pX (for X = 0; X <= 3)" on page 667	0x20000+ 0x2+(X*0x 100000)	PclkPtrInitVal_pX: PAC Hard Macro Command FIFO ReadPointer Initial Value

Register	Offset	Description
" <a href="#">CmdFifoWrModeMaster_pX (for X = 0; X &lt;= 3)" on page 668</a>	0x20000+0x3+(X*0x100000)	CmdFifoWrModeMaster_pX: PAC Hard Macro Command FIFO Write Mode Value
" <a href="#">MTestDtoCtrl" on page 669</a>	0x20000+0x4	Enables MTestCombo output on core-side copy
" <a href="#">PipeCtl_pX (for X = 0; X &lt;= 3)" on page 670</a>	0x20000+0x5+(X*0x100000)	Delay Enables for PIPE block
" <a href="#">LpDqPhaseDisable" on page 671</a>	0x20000+0x6	LpDqPhaseDisable: Disables the PAC hard macro clock when in LP state with clocks stopped
" <a href="#">PubDbyteDisable" on page 672</a>	0x20000+0x7	PubDbyteDisable: Forces PUB logic for a dbyte into low-power/disabled mode
" <a href="#">CPclkDivRatio_pX (for X = 0; X &lt;= 3)" on page 673</a>	0x20000+0xb+(X*0x100000)	Pclk clock divider ratios
" <a href="#">PipeNetDis" on page 675</a>	0x20000+0xc	PipeNetDis: Power Saving feature
" <a href="#">MiscPipeEn" on page 676</a>	0x20000+0xf	MiscPipeEn: Deprecated
" <a href="#">MtestMuxSelCMOS" on page 677</a>	0x20000+0x10	MtestMuxSelCMOS: Digital Observation Pin control for HMCMS
" <a href="#">EnRxDqsTracking_pX (for X = 0; X &lt;= 3)" on page 678</a>	0x20000+0x19+(X*0x100000)	Mode controls for fine RxEn timing adjustment during mission-mode operation.
" <a href="#">MtestMuxSel" on page 680</a>	0x20000+0x1a	MtestMuxSel: Digital Observation Pin control
" <a href="#">MASTERParityInvert" on page 681</a>	0x20000+0x4d	MASTERParityInvert: Invert APB Parity for register slave PUB_PAC
" <a href="#">DfiMode" on page 682</a>	0x20000+0x51	Enables for update and low-power interfaces for DFI0 and DFI1
" <a href="#">MtestPgmlInfo" on page 683</a>	0x20000+0x52	MtestPgmlInfo: Digital Observation Pin program info for debug
" <a href="#">PhyTID" on page 684</a>	0x20000+0x55	PhyTID: PHY Technology ID Register
" <a href="#">DbyteRxEnTrain" on page 685</a>	0x20000+0x59	DbyteRxEnTrain: Reserved for PHY training firmware use.
" <a href="#">PUBMODE" on page 686</a>	0x20000+0x6e	PUBMODE - HWT Mux Select

Register	Offset	Description
"DIITrainParam_pX (for X = 0; X <= 3)" on page 687	0x20000+0x71+(X*0x100000)	DLL Various Training Parameters
"DIIControl" on page 688	0x20000+0x78	DLL Lock State machine control register
"PulseDIIUpdatePhase" on page 689	0x20000+0x79	PulseDIIUpdatePhase: Reserved for PHY training firmware use.
"ScratchPadMASTER" on page 690	0x20000+0x7d	ScratchPadMASTER: ScratchPad for PUB_PAC
"PState" on page 691	0x20000+0x8b	PState: PSTATE Selection
"RxFifoInit" on page 692	0x20000+0xa0	Rx FIFO pointer initialization control
"ClockingCtrl" on page 693	0x20000+0xa2	ClockingCtrl: This register is dynamically written by PHY Initialization Engine during frequency...
"PhyPipeConfig" on page 694	0x20000+0xa3	Read Only displays PHY PIPE Configuration.
"PhyConfig" on page 696	0x20000+0xa4	Read Only displays PHY Configuration.
"LP5Mode" on page 699	0x20000+0xa5	LP5Mode: Selects LP5 Protocol
"ForceClkGaterEnables" on page 700	0x20000+0xa6	Forces internal clock gaters.
"D5Mode" on page 702	0x20000+0xa9	D5Mode: Deprecated
"PHYREV" on page 703	0x20000+0xee	PHYREV: The hardware version of this PHY, excluding the PUB
"TxRdPtrInit" on page 704	0x20000+0xf8	TxRdPtrInit: TxRdPtrInit control register
"MASTERReservedX (for X == 0)" on page 705	0x20000+0xfb	MASTERReservedX: Reserved for future use
"PUBReservedP1_pX (MASTER) (for X = 0; X <= 3)" on page 706	0x20000+0xff+(X*0x100000)	PUBReservedP1_pX: Reserved for future use
"PclkGateControl" on page 707	0x20000+0x200	Pclk gating override control

Register	Offset	Description
DWC_DDRPHYA_AC0_p0 (for j = 0; j <= DWC_LPDDR5XPHY_NUM_CHANNELS-1)(for k = 0; k <= 3) Exists: DWC_LPDDR5XPHY_NUM_CHANNELS > j	DWC_DDRPHYA_ACj_Pk (0x30000+(j<<12))+0x8+(X*0x100000)	DesignWare Cores LPDDR5X/54X PHY address block
"AcPipeEn_pX (for X = 0; X <= 3)" on page 710	(0x30000+(j<<12))+0xa	AcPipeEn_pX: Enable DfiClk Pipeline on the Command Address.
"CKDIIStopCal" on page 711	(0x30000+(j<<12))+0xa	CKDIIStopCal: Stop MDLL Calibration for incoming TxRdPtrInit
"CkVal_pX (for X = 0; X <= 3)" on page 712	(0x30000+(j<<12))+0xe+(X*0x100000)	CkVal_pX: Active clock encoding register for MEMCLK
"CkDisVal_pX (for X = 0; X <= 3)" on page 713	(0x30000+(j<<12))+0xf+(X*0x100000)	CkDisVal_pX: Mode select register for MEMCLK
"DTOBypassEn" on page 714	(0x30000+(j<<12))+0x10	DTOBypassEn: Async Flyover Tx Mode control for DTO
"ACSingleEndedMode_pX (for X = 0; X <= 3)" on page 715	(0x30000+(j<<12))+0x15+(X*0x100000)	Control for Single Ended LP4X/5 Mode
"InitSeqControl" on page 716	(0x30000+(j<<12))+0x16	AC/DBYTE Pipeline Init Sequence/Pipeline Control register
"MtestMuxSel" on page 720	(0x30000+(j<<12))+0x1a	MtestMuxSel: Digital Observation Pin control
"PorControl" on page 721	(0x30000+(j<<12))+0x20	PMU Power-on Reset Control
"ClrPORMemReset" on page 722	(0x30000+(j<<12))+0x21	ClrPORMemReset: Switch the POR Self-Initialized LATCH to 1
"MemResetL" on page 723	(0x30000+(j<<12))+0x22	Protection and control of BP_MEMRESET_L

Register	Offset	Description
"CMOSxHardMacroModeSel" on page 724	(0x30000+(j<<12))+0x24	CMOSxHardMacroModeSel: This CSR is deprecated
"RxAcVrefControl_pX (for X = 0; X <= 3)" on page 725	(0x30000+(j<<12))+0x25+(X*0x10000)	RxAcVref DAC control.
"MemResetLStatus" on page 726	(0x30000+(j<<12))+0x26	This Register provides BP_MEMRESET_L pad status
"ACDlyScaleGatingDisable" on page 727	(0x30000+(j<<12))+0x27	ACDlyScaleGatingDisable: Reserved for PHY training firmware use.
"LcdlCalControl" on page 728	(0x30000+(j<<12))+0x47	DLL Lock State machine control register
"ACParityInvert" on page 729	(0x30000+(j<<12))+0x4d	ACParityInvert: Invert APB Parity for register slave AC
"ACPulseDIIUpdatePhase" on page 730	(0x30000+(j<<12))+0x79	ACPulseDIIUpdatePhase: Reserved for PHY training firmware use.
"ScratchPadAC" on page 731	(0x30000+(j<<12))+0x7d	ScratchPadAC: ScratchPad for AC
"ATestMode" on page 732	(0x30000+(j<<12))+0x7f	ATestMode control for SE/Diff Address/Command slices
"AcDigStrobeGenSel" on page 734	(0x30000+(j<<12))+0x86	AcDigStrobeGenSel: Selects the source for the generation of the RxDigStrobe
"AcDigStrobePat" on page 735	(0x30000+(j<<12))+0x87	AcDigStrobePat: Selects the pattern for RxDigStrobe
"AcOdtEn" on page 736	(0x30000+(j<<12))+0x88	AcOdtEn: Per Slice in AC OdtEn Control
"AcRxStrobeEnPat" on page 737	(0x30000+(j<<12))+0x89	AcRxStrobeEnPat: Selects the pattern for RxStrobeEn for AC DIFF Slice.

Register	Offset	Description
"ATxDlySelect_pX (for X = 0; X <= 3)" on page 738	(0x30000+(j<<12))+0x8f+(X*0x100000)	ATxDlySelect_pX: Address/Command Delay Select
"MapCA0toDfi" on page 739	(0x30000+(j<<12))+0x90	MapCA0toDfi: Maps AC CH PHY CA lane 0 from dfi_address
"MapCA1toDfi" on page 740	(0x30000+(j<<12))+0x91	MapCA1toDfi: Maps AC CH PHY CA lane 1 from dfi_address
"MapCA2toDfi" on page 741	(0x30000+(j<<12))+0x92	MapCA2toDfi: Maps AC CH PHY CA lane 2 from dfi_address
"MapCA3toDfi" on page 742	(0x30000+(j<<12))+0x93	MapCA3toDfi: Maps AC CH PHY CA lane 3 from dfi_address
"MapCA4toDfi" on page 743	(0x30000+(j<<12))+0x94	MapCA4toDfi: Maps AC CH PHY CA lane 4 from dfi_address
"MapCA5toDfi" on page 744	(0x30000+(j<<12))+0x95	MapCA5toDfi: Maps AC CH PHY CA lane 5 from dfi_address
"MapCA6toDfi" on page 745	(0x30000+(j<<12))+0x96	MapCA6toDfi: Maps AC CH PHY CA lane 6 from dfi_address
"AsyncAcTxMode" on page 746	(0x30000+(j<<12))+0xa0	AsyncAcTxMode: Reserved for PHY training firmware use.
"AsyncAcRxMode" on page 747	(0x30000+(j<<12))+0xa1	AsyncAcRxMode: Reserved for PHY training firmware use.
"AsyncAcTxEn" on page 748	(0x30000+(j<<12))+0xa2	AsyncAcTxEn: Reserved for PHY training firmware use.
"AsyncAcTxData" on page 749	(0x30000+(j<<12))+0xa3	AsyncAcTxData: Reserved for PHY training firmware use.
"AsyncAcRxData" on page 750	(0x30000+(j<<12))+0xa5	AsyncAcRxData: Reserved for PHY training firmware use.

Register	Offset	Description
"ForceClkDisable" on page 751	(0x30000+(j<<12))+0xa6	ForceClkDisable: Clock gating control
"CaBusTriEn" on page 752	(0x30000+(j<<12))+0xab	CaBusTriEn: Enable Tri-Starting of the CA bus
"AcLnDisable" on page 753	(0x30000+(j<<12))+0xac	AcLnDisable: Per AC Lane Control to disable the lane
"DfiClkAcLnDis" on page 754	(0x30000+(j<<12))+0xad	DfiClkAcLnDis: Per AC Lane Control to disable the DfiClk in the lane
"PClkAcLnDis" on page 755	(0x30000+(j<<12))+0xae	PClkAcLnDis: Per AC Lane Control to disable the PClk in the lane
"AcLcdlCalPhDetOut" on page 756	(0x30000+(j<<12))+0xaf	AcLcdlCalPhDetOut: Reserved for PHY training firmware use.
"ACXTxDly_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 9)" on page 757	(0x30000+(j<<12))+0xd8+(X*0x100000)+(Y*0x100)	ACXTxDly_rY_pX: Address/Command Delay, per pstate.
"CKXTxDly_pX (for X = 0; X <= 3)" on page 758	(0x30000+(j<<12))+0xd9+(X*0x100000)	CKXTxDly_pX: Reserved for PHY training firmware use.
"ACXTxDlyDTO_pX (for X = 0; X <= 3)" on page 759	(0x30000+(j<<12))+0xda+(X*0x100000)	ACXTxDlyDTO_pX: Address/Command Delay, per pstate.
"ACXTxDly2nMode_rY_pX (for X = 0; X <= 3)(for Y = 0; Y <= 9)" on page 760	(0x30000+(j<<12))+0xde+(X*0x100000)+(Y*0x100)	ACXTxDly2nMode_rY_pX: Unused for LP5 mode.
"AcLcdlUpdInterval_pX (for X = 0; X <= 3)" on page 761	(0x30000+(j<<12))+0xeb+(X*0x100000)	AcLcdlUpdInterval_pX: Controls for the clock ACX2 DLLs

Register	Offset	Description
"LcdlCalSeqUpdCK_pX (for X = 0; X <= 3)" on page 762	(0x30000+(j<<12))+0 xec+(X*0x100000)	LcdlCalSeqUpdCK_pX: Controls the LCDLCALSEQ_CK LongBubble
"CkTriEn" on page 763	(0x30000+(j<<12))+0 xed	CkTriEn: Enables the AC CK to Tristate during dram_clk_disable
"ACReservedX (for X == 0)" on page 764	(0x30000+(j<<12))+0 xf9	ACReservedX: Reserved for future use
"LcdlCalCtrl" on page 765	(0x30000+(j<<12))+0 xfe	LcdlCalCtrl: Reserved for PHY test firmware use.
"PUBReservedP1_pX (AC) (for X = 0; X <= 3)" on page 766	(0x30000+(j<<12))+0 xff+(X*0x100000)	PUBReservedP1_pX: Reserved for future use
"PclkDCDCtrl_pX (AC) (for X = 0; X <= 3)" on page 767	(0x30000+(j<<12))+0 x100+(X*0x100000)	PclkDCDCtrl_pX: Controls the DCD Comparator for DCA calibration
"ForceInternalUpdate" on page 768	(0x30000+(j<<12))+0 x103	ForceInternalUpdate: This Register used by Training Firmware to force an internal PHY Update...
"ATestPrbsErrCntSECX (for X = 0; X <= 1)" on page 769	(0x30000+(j<<12))+0 x1a8+(X*0x1)	ATestPrbsErrCntSECX: Address Loopback Test Result register
"AcPDsampleSECX (for X = 0; X <= 1)" on page 770	(0x30000+(j<<12))+0 x1b8+(X*0x1)	AcPDsampleSECX: Address Loopback sample values for SEC slice
"AcPDsampleDIFF" on page 771	(0x30000+(j<<12))+0 x1ca	Address Loopback sample values for Diff slice
"ATestPrbsErrCntDIFF0T" on page 772	(0x30000+(j<<12))+0 x3e0	ATestPrbsErrCntDIFF0T: Address Loopback Test Result register
"ATestPrbsErrCntDIFF0C" on page 773	(0x30000+(j<<12))+0 x3e4	ATestPrbsErrCntDIFF0C: Address Loopback Test Result register

Register	Offset	Description
"PclkDCACalCtrl0AC" on page 774	(0x30000+(j<<12))+0 x800	Controls DCA sample polarity
"PclkDCADynCtrl" on page 775	(0x30000+(j<<12))+0 x802	Dynamic bits for DCA control
"PclkDCASStaticCtrl0AC_pX (for X = 0; X <= 3)" on page 776	(0x30000+(j<<12))+0 x803+(X*0 x100000)	Static bits for DCA control in AC and CK blocks
"PclkDCASampCntAC" on page 778	(0x30000+(j<<12))+0 x804	Sample counts for various stages of DCA calibration
"PclkDCAHysMaskAC" on page 779	(0x30000+(j<<12))+0 x805	PclkDCAHysMaskAC: Used by DCA quick search algorithm
"PclkDCACalFineBoundAC" on page 780	(0x30000+(j<<12))+0 x806	Sets limit values of DCA Fine searches
"PclkDCANextFineOnCoarseAC" on page 781	(0x30000+(j<<12))+0 x807	Sets the DCAFine value to use when Coarse changes
"PclkDCAFullSearchIVACAC" on page 782	(0x30000+(j<<12))+0 x808	Initial DCA Fine value to use for full searches
"LcdITstCtrl" on page 783	(0x30000+(j<<12))+0 x884	LcdITstCtrl: Reserved for PHY test firmware use.
"AcLoopBackEnLnX (for X = 0; X <= 10)" on page 784	(0x30000+(j<<12))+0 x900+(X*0 x1)	AcLoopBackEnLnX: Per AC Lane, Loopback Enable Configuration
"ATestPrbsErrCntSEX (for X = 0; X <= 7)" on page 785	(0x30000+(j<<12))+0 xb00+(X*0 x1)	ATestPrbsErrCntSEX: Address Loopback Test Result register
DWC_DDRPHYA_PPGC0_p0 DesignWare Cores LPDDR5X/5/4X PHY address block DWC_DDRPHYA_PPGCj_Pk (for j == 0)(for k = 0; k <= 3) Exists: Always		

Register	Offset	Description
" <a href="#">PpgcGenCtrl</a> " on page <a href="#">787</a>	0x70000+0x0	PpgcGenCtrl: Reserved for PHY training firmware use.
" <a href="#">PpgcGenDbiCtrl</a> " on page <a href="#">788</a>	0x70000+0x1	PpgcGenDbiCtrl: Reserved for PHY training firmware use.
" <a href="#">PpgcGenDbiConfig</a> " on page <a href="#">789</a>	0x70000+0x2	PpgcGenDbiConfig: Reserved for PHY training firmware use.
" <a href="#">PpgcGenLaneMuxSelX (for X = 0; X &lt;= 1)</a> " on page <a href="#">790</a>	0x70000+0x3+(X*0x1)	PpgcGenLaneMuxSelX: Reserved for PHY training firmware use.
" <a href="#">EnPhyUpdZQCalUpdate</a> " on page <a href="#">791</a>	0x70000+0x5	EnPhyUpdZQCalUpdate: Enable ZQ Cal on Phy Update
" <a href="#">BlockDfilInterface</a> " on page <a href="#">792</a>	0x70000+0x6	BlockDfilInterface: Used to block certain operations when PState info is being transferred from...
" <a href="#">BlockDfilInterfaceStatus</a> " on page <a href="#">793</a>	0x70000+0x7	BlockDfilInterfaceStatus: Tracks blocked assertions of sideband signals when...
" <a href="#">DfiCustMode_pX (for X = 0; X &lt;= 3)</a> " on page <a href="#">794</a>	0x70000+0xb+(X*0x100000)	DfiCustMode_pX: Deprecated
" <a href="#">HwtMRL_pX (for X = 0; X &lt;= 3)</a> " on page <a href="#">795</a>	0x70000+0xd+(X*0x100000)	HwtMRL_pX: HWT MaxReadLatency.
" <a href="#">RegRet</a> " on page <a href="#">796</a>	0x70000+0xe	RegRet: Reserved for PHY training firmware use.
" <a href="#">DisableZQupdateOnSnoop</a> " on page <a href="#">797</a>	0x70000+0xf	DisableZQupdateOnSnoop: Disable ZQUpdate (part of ctrlupd) if snoop_osc_running asserted
" <a href="#">Prbs0GenModeSel</a> " on page <a href="#">798</a>	0x70000+0x10	Prbs0GenModeSel: Reserved for PHY training firmware use.
" <a href="#">Prbs0GenUiMuxSel</a> " on page <a href="#">799</a>	0x70000+0x11	Prbs0GenUiMuxSel: Reserved for PHY training firmware use.
" <a href="#">Prbs0GenTapDlyX (for X = 0; X &lt;= 7)</a> " on page <a href="#">800</a>	0x70000+0x12+(X*0x1)	Prbs0GenTapDlyX: Reserved for PHY training firmware use.
" <a href="#">MtestMuxSel</a> " on page <a href="#">801</a>	0x70000+0x1a	MtestMuxSel: Digital Observation Pin control
" <a href="#">Prbs0GenStateLo</a> " on page <a href="#">802</a>	0x70000+0x1b	Prbs0GenStateLo: Reserved for PHY training firmware use.
" <a href="#">Prbs0GenStateHi</a> " on page <a href="#">803</a>	0x70000+0x1c	Prbs0GenStateHi: Reserved for PHY training firmware use.

Register	Offset	Description
"Prbs1GenModeSel" on page 804	0x70000+0x20	Prbs1GenModeSel: Reserved for PHY training firmware use.
"Prbs1GenUiMuxSel" on page 805	0x70000+0x21	Prbs1GenUiMuxSel: Reserved for PHY training firmware use.
"Prbs1GenTapDlyX (for X = 0; X <= 7)" on page 806	0x70000+0x22+(X*0x1)	Prbs1GenTapDlyX: Reserved for PHY training firmware use.
"Prbs1GenStateLo" on page 807	0x70000+0x2b	Prbs1GenStateLo: Reserved for PHY training firmware use.
"Prbs1GenStateHi" on page 808	0x70000+0x2c	Prbs1GenStateHi: Reserved for PHY training firmware use.
"Prbs2GenModeSel" on page 809	0x70000+0x30	Prbs2GenModeSel: Reserved for PHY training firmware use.
"Prbs2GenUiMuxSel" on page 810	0x70000+0x31	Prbs2GenUiMuxSel: Reserved for PHY training firmware use.
"Prbs2GenTapDlyX (for X = 0; X <= 7)" on page 811	0x70000+0x32+(X*0x1)	Prbs2GenTapDlyX: Reserved for PHY training firmware use.
"Prbs2GenStateLo" on page 812	0x70000+0x3b	Prbs2GenStateLo: Reserved for PHY training firmware use.
"Prbs2GenStateHi" on page 813	0x70000+0x3c	Prbs2GenStateHi: Reserved for PHY training firmware use.
"PPTTrainSetup_pX (for X = 0; X <= 3)" on page 814	0x70000+0x40+(X*0x100000)	Setup Intervals for DFI PHY Master operations
"PhyMstrFreqOverride_pX (for X = 0; X <= 3)" on page 816	0x70000+0x41+(X*0x100000)	PhyMstrFreqOverride_pX: Programmed by PHY training firmware to support LPDDR4X DRAM drift...
"DfilnInitComplete" on page 817	0x70000+0x49	DfilnInitComplete: DFI Init Complete control
"PPGCParityInvert" on page 818	0x70000+0x4d	PPGCParityInvert: Invert APB Parity for register slave PPGC
"PMIEnable" on page 819	0x70000+0x54	PMIEnable: This register is dynamically written by PHY Initialization Engine during frequency changes...
"Dfi0Status" on page 820	0x70000+0x5a	Dfi0Status: Current state of certain Dfi Inputs
"Dfi1Status" on page 821	0x70000+0x5b	Dfi1Status: Current state of certain Dfi Inputs

Register	Offset	Description
"DfiHandshakeDelays0_pX (for X = 0; X <= 3)" on page <a href="#">822</a>	0x70000+0x66+(X*0x100000)	Small delays on handshake signals per AC Channel
"DFIPHYUPDX (for X = 0; X <= 1)" on page <a href="#">823</a>	0x70000+0x67+(X*0x80)	Per AC Ch DFIPhyUpd Request time cntr (in DfiClk)
"DfiLpCtrlEnX (for X = 0; X <= 1)" on page <a href="#">825</a>	0x70000+0x68+(X*0x80)	DfiLpCtrlEnX: DFI LP Ctrl Request power saving controls.
"DfiLpDataEnX (for X = 0; X <= 1)" on page <a href="#">826</a>	0x70000+0x69+(X*0x80)	DfiLpDataEnX: DFI LP Data Request power saving controls.
"DynOdtEnCntrlX (for X = 0; X <= 1)" on page <a href="#">827</a>	0x70000+0x6a+(X*0x80)	Dynamically Disable the ODT during low power
"DfiRespHandshakeDelays0_pX (for X = 0; X <= 3)" on page <a href="#">828</a>	0x70000+0x6b+(X*0x100000)	Small delays on handshake signals per AC Channel
"HwtLpCsEnA" on page <a href="#">829</a>	0x70000+0x72	HwtLpCsEnA: Programmed by PHY training firmware.
"HwtLpCsEnB" on page <a href="#">830</a>	0x70000+0x73	HwtLpCsEnB: Programmed by PHY training firmware.
"HwtCtrl" on page <a href="#">831</a>	0x70000+0x77	HwtCtrl: Programmed by PHY training firmware to support LPDDR4X/5 DRAM drift compensation.
"HwtControlOvr" on page <a href="#">832</a>	0x70000+0x7a	HwtControlOvr: Reserved for PHY training firmware use.
"ScratchPadPPGC" on page <a href="#">833</a>	0x70000+0x7d	ScratchPadPPGC: ScratchPad for PPGC
"HwtControlVal" on page <a href="#">834</a>	0x70000+0x7e	HwtControlVal: Reserved for PHY training firmware use.
"ForceHWTClkGaterEnables" on page <a href="#">835</a>	0x70000+0x80	Forces internal clock gaters.
"MasUpdGoodCtr" on page <a href="#">836</a>	0x70000+0xb5	MasUpdGoodCtr: Counts successful PHY Master Interface Updates (PPTs)
"PhyUpd0GoodCtr" on page <a href="#">837</a>	0x70000+0xb6	PhyUpd0GoodCtr: Counts acknowledged PHY-initiated DFI0 Interface Updates
"PhyUpd1GoodCtr" on page <a href="#">838</a>	0x70000+0xb7	PhyUpd1GoodCtr: Counts acknowledged PHY-initiated DFI1 Interface Updates

Register	Offset	Description
"CtlUpd0GoodCtr" on page 839	0x70000+0xb8	CtlUpd0GoodCtr: Counts acknowledged Memory Controller DFI0 Interface Updates
"CtlUpd1GoodCtr" on page 840	0x70000+0xb9	CtlUpd1GoodCtr: Counts acknowledged Memory Controller DFI1 Interface Updates
"MasUpdFailCtr" on page 841	0x70000+0xba	MasUpdFailCtr: Counts unsuccessful PHY Master Interface Updates
"PhyUpd0FailCtr" on page 842	0x70000+0xbb	PhyUpd0FailCtr: Counts unsuccessful PHY-initiated DFI0 Interface Updates
"PhyUpd1FailCtr" on page 843	0x70000+0xbc	PhyUpd1FailCtr: Counts unsuccessful PHY-initiated DFI1 Interface Updates
"PhyPerfCtrEnable" on page 844	0x70000+0xbd	Enables for Performance Counters
"DfiHandshakeDelays1_pX (for X = 0; X <= 3)" on page 846	0x70000+0xe6+(X*0x100000)	Small delays on handshake signals per AC Channel
"DfiRespHandshakeDelays1_pX (for X = 0; X <= 3)" on page 847	0x70000+0xeb+(X*0x100000)	Small delays on handshake signals per AC Channel
"FspSkipList" on page 848	0x70000+0xf0	FspState update skip list.
"PPGCReservedX (for X == 0)" on page 849	0x70000+0xfa	PPGCReservedX: Reserved for future use
"PUBReservedP1_pX (PPGC) (for X = 0; X <= 3)" on page 850	0x70000+0xff+(X*0x100000)	PUBReservedP1_pX: Reserved for future use
"PhyInterruptOverride" on page 851	0x70000+0x11a	PhyInterruptOverride: Interrupt Override Debug Register
"PhyInterruptEnable" on page 852	0x70000+0x11b	Interrupt Enable Bits
"PhyInterruptFWControl" on page 854	0x70000+0x11c	Interrupt Firmware Control Bits
"PhyInterruptMask" on page 856	0x70000+0x11d	Interrupt Mask Bits
"PhyInterruptClear" on page 858	0x70000+0x11e	Interrupt Clear Bits
"PhyInterruptStatus" on page 860	0x70000+0x11f	Interrupt Status Bits

Register	Offset	Description
"ACSMRunCtrl" on page 863	0x70000+0x120	Enable for ACSMs and optionally specifies the AcsmProgPtr.
"ACSMDone" on page 865	0x70000+0x121	ACSMDone: Done Status for ACSMs
"ACSMStartAddr_pX (for X = 0; X <= 3)" on page 866	0x70000+0x122+(X*0x100000)	ACSMStartAddr_pX: Start Address for ACSM Sequence
"ACSMStopAddr_pX (for X = 0; X <= 3)" on page 867	0x70000+0x123+(X*0x100000)	ACSMStopAddr_pX: Last address of Playback Sequence
"ACSMLastAddr" on page 868	0x70000+0x124	ACSMLastAddr: Last executed address of Playback Sequence
"ACSMAlgIncVal" on page 869	0x70000+0x125	ACSMAlgIncVal: Increment value for ACSM Algorithmic Address Generator
"ACSMAddressMask" on page 870	0x70000+0x126	ACSMAddressMask: CA Address Mask
"ACSMOuterLoopRepeatCnt" on page 871	0x70000+0x127	ACSMOuterLoopRepeatCnt: Outer Loop Repeat
"ACSMCkeControl" on page 872	0x70000+0x128	ACSMCkeControl: CKE Control for ACSM
"ACSMCkeStatus" on page 873	0x70000+0x129	ACSMCkeStatus: CKE Status for ACSM
"ACSMWckEnControl" on page 874	0x70000+0x12a	ACSMWckEnControl: hwt_wck_en Control for ACSM
"ACSMWckEnStatus" on page 875	0x70000+0x12b	ACSMWckEnStatus: wck Status for ACSM
"ACSMRxEnPulse_pX (for X = 0; X <= 3)" on page 876	0x70000+0x12c+(X*0x100000)	Timing Parameters for RxEn Pulse
"ACSMRxValPulse_pX (for X = 0; X <= 3)" on page 877	0x70000+0x12d+(X*0x100000)	Timing Parameters for RxVal Pulse
"ACSMTxEnPulse_pX (for X = 0; X <= 3)" on page 878	0x70000+0x12e+(X*0x100000)	Timing Parameters for TxEn Pulse
"ACSMWrcsPulse_pX (for X = 0; X <= 3)" on page 879	0x70000+0x12f+(X*0x100000)	Timing Parameters for Wrcs Pulse

Register	Offset	Description
"ACSMRdcsPulse_pX (for X = 0; X <= 3)" on page 880	0x70000+0x130+(X*0x100000)	Timing Parameters for Rdcs Pulse
"ACSMInfiniteOLRC" on page 881	0x70000+0x131	ACSMInfiniteOLRC: Enables outer loop as infinite for ACSMs
"ACSMDefaultAddr" on page 882	0x70000+0x132	ACSMDefaultAddr: ca value driven by ACSM when idle
"ACSMDefaultCs" on page 883	0x70000+0x133	ACSMDefaultCs: cs value driven by ACSM when idle
"ACSMStaticCtrl" on page 884	0x70000+0x134	Control the ACSM Phase mode of operation.
"ACSMWckWriteStaticLoPulse_pX (for X = 0; X <= 3)" on page 885	0x70000+0x135+(X*0x100000)	Timing Parameters for WckWriteStaticLo Pulse
"ACSMWckWriteStaticHiPulse_pX (for X = 0; X <= 3)" on page 886	0x70000+0x136+(X*0x100000)	Timing Parameters for WckWriteStaticHi Pulse
"ACSMWckWriteTogglePulse_pX (for X = 0; X <= 3)" on page 887	0x70000+0x137+(X*0x100000)	Timing Parameters for WckWriteToggle Pulse
"ACSMWckWriteFastTogglePulse_pX (for X = 0; X <= 3)" on page 888	0x70000+0x138+(X*0x100000)	Timing Parameters for WckWriteFastToggle Pulse
"ACSMWckReadStaticLoPulse_pX (for X = 0; X <= 3)" on page 889	0x70000+0x139+(X*0x100000)	Timing Parameters for WckReadStaticLo Pulse
"ACSMWckReadStaticHiPulse_pX (for X = 0; X <= 3)" on page 890	0x70000+0x13a+(X*0x100000)	Timing Parameters for WckReadStaticHi Pulse
"ACSMWckReadTogglePulse_pX (for X = 0; X <= 3)" on page 891	0x70000+0x13b+(X*0x100000)	Timing Parameters for WckReadToggle Pulse
"ACSMWckReadFastTogglePulse_pX (for X = 0; X <= 3)" on page 892	0x70000+0x13c+(X*0x100000)	Timing Parameters for WckReadFastToggle Pulse
"ACSMWckFreqSwStaticLoPulse_pX (for X = 0; X <= 3)" on page 893	0x70000+0x13d+(X*0x100000)	Timing Parameters for WckFreqSwStaticLo Pulse

Register	Offset	Description
"ACSMWckFreqSwStaticHiPulse_pX (for X = 0; X <= 3)" on page 894	0x70000+0x13e+(X*0x100000)	Timing Parameters for WckFreqSwStaticHi Pulse
"ACSMWckFreqSwTogglePulse_pX (for X = 0; X <= 3)" on page 895	0x70000+0x13f+(X*0x100000)	Timing Parameters for WckFreqSwToggle Pulse
"ACSMWckFreqSwFastTogglePulse_pX (for X = 0; X <= 3)" on page 896	0x70000+0x140+(X*0x100000)	Timing Parameters for WckFreqSwFastToggle Pulse
"ACSMWckFreeRunMode_pX (for X = 0; X <= 3)" on page 898	0x70000+0x141+(X*0x100000)	ACSMWckFreeRunMode_pX: Extends wck pulses in free running mode
"ACSMLowSpeedClockEnable" on page 899	0x70000+0x142	ACSMLowSpeedClockEnable: Used to control and reduce speed of ACSM Output and AC bump clocks
"ACSMLowSpeedClockDelay" on page 900	0x70000+0x144	ACSMLowSpeedClockDelay: Used to offset reduced speed AC bump clocks
"ACSMRptCntOverride_pX (for X = 0; X <= 3)" on page 901	0x70000+0x145+(X*0x100000)	ACSMRptCntOverride_pX: Used to override the REPEATCNT field of the ACSM Control Word
"ACSMRptCntDbl_pX (for X = 0; X <= 3)" on page 902	0x70000+0x146+(X*0x100000)	ACSMRptCntDbl_pX: Used to double the REPEATCNT field of the ACSM Control Word
"ACSMParityStatus" on page 903	0x70000+0x147	ACSMParityStatus: ACSM Byte Parity Error Status
"HwtLpCsEnBypass" on page 904	0x70000+0x174	HwtLpCsEnBypass: Reserved for PHY training firmware use.
"ACSMNopAddr" on page 905	0x70000+0x18a	ACSMNopAddr: Enable ACSMXlat address dereferencing
"SnoopCntrl" on page 906	0x70000+0x1a7	SnoopCntrl: Selects the value to be driven on hwt_snoop_en in HWT mode.
"ACSMParityInvert" on page 907	0x70000+0x1a8	ACSMParityInvert: Forces parity errors in ACSM RAM
"AcsmPsIndx" on page 908	0x70000+0x1a9	AcsmPsIndx: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"AcsmDynPtrCtrl" on page 909	0x70000+0x1aa	AcsmDynPtrCtrl: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency...

Register	Offset	Description
"FspState" on page 910	0x70000+0x1ef	Captures the state of status interface for retention enter/exit
"AcsmMapTableX (for X = 0; X <= 63)" on page 911	0x70000+0x200+(X*0x1)	AcsmMapTableX: Maps ProgPtr+DynPtr into XlatTable Address
"AcsmStartAddrXlatValX (for X = 0; X <= 63)" on page 912	0x70000+0x324+(X*0x1)	AcsmStartAddrXlatValX: Start Address value of AcsmPtr Translation Register0
"AcsmStopAddrXlatValX (for X = 0; X <= 63)" on page 913	0x70000+0x38b+(X*0x1)	AcsmStopAddrXlatValX: Stop Address value of AcsmPtr Translation Register0
DWC_DDRPHYA_INITENG0_p0 DesignWare Cores LPDDR5X/5/4X PHY address block DWC_DDRPHYA_INITENGj_Pk (for j == 0)(for k = 0; k <= 3) Exists: Always		
"DVFSCEn_pX (for X = 0; X <= 3)" on page 915	0x90000+0x10+(X*0x100000)	DVFSCEn_pX: Deprecate
"MtestMuxSel" on page 916	0x90000+0x1a	MtestMuxSel: Digital Observation Pin control
"StartVector0bX (for X = 0; X <= 15)" on page 917	0x90000+0x1c+(X*0x1)	StartVector0bX: Programmed by dwc_ddrphy_phyinit_l_loadPIEImage() to support frequency...
"PhyInLPX (for X = 0; X <= 3)" on page 918	0x90000+0x2d	PhyInLPX: Indicator for PIE Lower Power 3 (LP3) Status
"INITENGParityInvert" on page 919	0x90000+0x4d	INITENGParityInvert: Invert APB Parity for register slave INITENG
"ScratchPadINITENG" on page 920	0x90000+0x7d	ScratchPadINITENG: ScratchPad for INITENG
"PieCtrlStartVec0_pX (for X = 0; X <= 3)" on page 921	0x90000+0x708+(X*0x100000)	PieCtrlStartVec0_pX: Programmed by dwc_ddrphy_phyinit_l_loadPIEImage() to support frequency...
"PIEProgErrStatus" on page 922	0x90000+0x709	PIE Programming Error Status
"PIEParityInvert" on page 924	0x90000+0x70a	PIEParityInvert: Forces parity errors in PIE RAM
"PIEParityStatus" on page 925	0x90000+0x70b	PIEParityStatus: PIE Byte Parity Error Status

Register	Offset	Description
<a href="#">“Seq0BDisableFlagX (for X = 0; X &lt;= 31)” on page 926</a>	0x90000+ 0x70c+(X* 0x1)	Seq0BDisableFlagX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency...
<a href="#">“Seq0BCSRUpperWrData” on page 927</a>	0x90000+ 0x71c	Seq0BCSRUpperWrData: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency...
<a href="#">“Seq0BCSRUpperRdData” on page 928</a>	0x90000+ 0x71d	Seq0BCSRUpperRdData: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency...
<a href="#">“WaitCondUC” on page 929</a>	0x90000+ 0x71e	WaitCondUC: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
<a href="#">“PhyMstrPMValOverride” on page 930</a>	0x90000+ 0x71f	PhyMstrPMValOverride: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency...
<a href="#">“PIEDebugStatus” on page 931</a>	0x90000+ 0x720	PIEDebugStatus: Assorted status signals to help with PIE debug
<a href="#">“DestPstate” on page 932</a>	0x90000+ 0x721	DestPstate: Read by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
<a href="#">“DfiFreqXlatDestPStateX (for X = 0; X &lt;= 15)” on page 933</a>	0x90000+ 0x730+(X* 0x1)	DFI Frequency Translation Register 0 for Dest PState
<a href="#">“Seq0BGPR0_pX (for X = 0; X &lt;= 3)” on page 934</a>	0x90000+ 0x800+(X* 0x100000)	Seq0BGPR0_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
<a href="#">“Seq0BGPR1_pX (for X = 0; X &lt;= 3)” on page 935</a>	0x90000+ 0x801+(X* 0x100000)	Seq0BGPR1_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
<a href="#">“Seq0BGPR2_pX (for X = 0; X &lt;= 3)” on page 936</a>	0x90000+ 0x802+(X* 0x100000)	Seq0BGPR2_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
<a href="#">“Seq0BGPR3_pX (for X = 0; X &lt;= 3)” on page 937</a>	0x90000+ 0x803+(X* 0x100000)	Seq0BGPR3_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
<a href="#">“Seq0BGPR4_pX (for X = 0; X &lt;= 3)” on page 938</a>	0x90000+ 0x804+(X* 0x100000)	Seq0BGPR4_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
<a href="#">“Seq0BGPR5_pX (for X = 0; X &lt;= 3)” on page 939</a>	0x90000+ 0x805+(X* 0x100000)	Seq0BGPR5_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.

Register	Offset	Description
"Seq0BGPR6_pX (for X = 0; X <= 3)" on page 940	0x90000+ 0x806+(X* 0x100000)	Seq0BGPR6_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR7_pX (for X = 0; X <= 3)" on page 941	0x90000+ 0x807+(X* 0x100000)	Seq0BGPR7_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR8_pX (for X = 0; X <= 3)" on page 942	0x90000+ 0x808+(X* 0x100000)	Seq0BGPR8_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR9_pX (for X = 0; X <= 3)" on page 943	0x90000+ 0x809+(X* 0x100000)	Seq0BGPR9_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR10_pX (for X = 0; X <= 3)" on page 944	0x90000+ 0x80a+(X* 0x100000)	Seq0BGPR10_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR11_pX (for X = 0; X <= 3)" on page 945	0x90000+ 0x80b+(X* 0x100000)	Seq0BGPR11_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR12_pX (for X = 0; X <= 3)" on page 946	0x90000+ 0x80c+(X* 0x100000)	Seq0BGPR12_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR13_pX (for X = 0; X <= 3)" on page 947	0x90000+ 0x80d+(X* 0x100000)	Seq0BGPR13_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR14_pX (for X = 0; X <= 3)" on page 948	0x90000+ 0x80e+(X* 0x100000)	Seq0BGPR14_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR15_pX (for X = 0; X <= 3)" on page 949	0x90000+ 0x80f+(X* 0x100000)	Seq0BGPR15_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR16_pX (for X = 0; X <= 3)" on page 950	0x90000+ 0x810+(X* 0x100000)	Seq0BGPR16_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR17_pX (for X = 0; X <= 3)" on page 951	0x90000+ 0x811+(X* 0x100000)	Seq0BGPR17_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR18_pX (for X = 0; X <= 3)" on page 952	0x90000+ 0x812+(X* 0x100000)	Seq0BGPR18_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.

Register	Offset	Description
"Seq0BGPR19_pX (for X = 0; X <= 3)" on page 953	0x90000+ 0x813+(X* 0x100000)	Seq0BGPR19_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR20_pX (for X = 0; X <= 3)" on page 954	0x90000+ 0x814+(X* 0x100000)	Seq0BGPR20_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR21_pX (for X = 0; X <= 3)" on page 955	0x90000+ 0x815+(X* 0x100000)	Seq0BGPR21_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR22_pX (for X = 0; X <= 3)" on page 956	0x90000+ 0x816+(X* 0x100000)	Seq0BGPR22_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR23_pX (for X = 0; X <= 3)" on page 957	0x90000+ 0x817+(X* 0x100000)	Seq0BGPR23_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR24_pX (for X = 0; X <= 3)" on page 958	0x90000+ 0x818+(X* 0x100000)	Seq0BGPR24_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR25_pX (for X = 0; X <= 3)" on page 959	0x90000+ 0x819+(X* 0x100000)	Seq0BGPR25_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR26_pX (for X = 0; X <= 3)" on page 960	0x90000+ 0x81a+(X* 0x100000)	Seq0BGPR26_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR27_pX (for X = 0; X <= 3)" on page 961	0x90000+ 0x81b+(X* 0x100000)	Seq0BGPR27_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR28_pX (for X = 0; X <= 3)" on page 962	0x90000+ 0x81c+(X* 0x100000)	Seq0BGPR28_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR29_pX (for X = 0; X <= 3)" on page 963	0x90000+ 0x81d+(X* 0x100000)	Seq0BGPR29_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR30_pX (for X = 0; X <= 3)" on page 964	0x90000+ 0x81e+(X* 0x100000)	Seq0BGPR30_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR31_pX (for X = 0; X <= 3)" on page 965	0x90000+ 0x81f+(X* 0x100000)	Seq0BGPR31_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.

Register	Offset	Description
"Seq0BGPRPage" on page 966	0x90000+0x828	Seq0BGPRPage: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BFixedAddrBits" on page 967	0x90000+0x829	Seq0BFixedAddrBits: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency...
"PieDynFlagMaskX (for X = 0; X <= 3)" on page 968	0x90000+0x82a+(X*0x1)	PieDynFlagMaskX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency...
"PieDynFlagSel" on page 969	0x90000+0x82e	PieDynFlagSel: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BPieGPRSel" on page 970	0x90000+0x82f	Seq0BPieGPRSel: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency...
"Seq0BDLY0_pX (for X = 0; X <= 3)" on page 971	0x90000+0x8e0+(X*0x100000)	Seq0BDLY0_pX: PHY Initialization Engine (PIE) Delay Register 0
"Seq0BDLY1_pX (for X = 0; X <= 3)" on page 972	0x90000+0x8e1+(X*0x100000)	Seq0BDLY1_pX: PHY Initialization Engine (PIE) Delay Register 1
"Seq0BDLY2_pX (for X = 0; X <= 3)" on page 973	0x90000+0x8e2+(X*0x100000)	Seq0BDLY2_pX: PHY Initialization Engine (PIE) Delay Register 2
"Seq0BDLY3_pX (for X = 0; X <= 3)" on page 974	0x90000+0x8e3+(X*0x100000)	Seq0BDLY3_pX: PHY Initialization Engine (PIE) Delay Register 3
"Seq0BDLY4_pX (for X = 0; X <= 3)" on page 975	0x90000+0x8e4+(X*0x100000)	Seq0BDLY4_pX: PHY Initialization Engine (PIE) Delay Register 4
"Seq0BDLY5_pX (for X = 0; X <= 3)" on page 976	0x90000+0x8e5+(X*0x100000)	Seq0BDLY5_pX: PHY Initialization Engine (PIE) Delay Register 5
"Seq0BDLY6_pX (for X = 0; X <= 3)" on page 977	0x90000+0x8e6+(X*0x100000)	Seq0BDLY6_pX: PHY Initialization Engine (PIE) Delay Register 6
"Seq0BDLY7_pX (for X = 0; X <= 3)" on page 978	0x90000+0x8e7+(X*0x100000)	Seq0BDLY7_pX: PHY Initialization Engine (PIE) Delay Register 7

Register	Offset	Description
"Seq0BDLY8_pX (for X = 0; X <= 3)" on page 979	0x90000+0x8e8+(X*0x100000)	Seq0BDLY8_pX: PHY Initialization Engine (PIE) Delay Register 8
"Seq0BDLY9_pX (for X = 0; X <= 3)" on page 980	0x90000+0x8e9+(X*0x100000)	Seq0BDLY9_pX: PHY Initialization Engine (PIE) Delay Register 9
"Seq0BDLY10_pX (for X = 0; X <= 3)" on page 981	0x90000+0x8ea+(X*0x100000)	Seq0BDLY10_pX: PHY Initialization Engine (PIE) Delay Register 10
"Seq0BDLY11_pX (for X = 0; X <= 3)" on page 982	0x90000+0x8eb+(X*0x100000)	Seq0BDLY11_pX: PHY Initialization Engine (PIE) Delay Register 11
"Seq0BDLY12_pX (for X = 0; X <= 3)" on page 983	0x90000+0x8ec+(X*0x100000)	Seq0BDLY12_pX: PHY Initialization Engine (PIE) Delay Register 12
"Seq0BDLY13_pX (for X = 0; X <= 3)" on page 984	0x90000+0x8ed+(X*0x100000)	Seq0BDLY13_pX: PHY Initialization Engine (PIE) Delay Register 13
"Seq0BDLY14_pX (for X = 0; X <= 3)" on page 985	0x90000+0x8ee+(X*0x100000)	Seq0BDLY14_pX: PHY Initialization Engine (PIE) Delay Register 14
"Seq0BDLY15_pX (for X = 0; X <= 3)" on page 986	0x90000+0x8ef+(X*0x100000)	Seq0BDLY15_pX: PHY Initialization Engine (PIE) Delay Register 15
"Seq0BDLY16_pX (for X = 0; X <= 3)" on page 987	0x90000+0x8f0+(X*0x100000)	Seq0BDLY16_pX: PHY Initialization Engine (PIE) Delay Register 16
"Seq0BDLY17_pX (for X = 0; X <= 3)" on page 988	0x90000+0x8f1+(X*0x100000)	Seq0BDLY17_pX: PHY Initialization Engine (PIE) Delay Register 17
"Seq0BDLY18_pX (for X = 0; X <= 3)" on page 989	0x90000+0x8f2+(X*0x100000)	Seq0BDLY18_pX: PHY Initialization Engine (PIE) Delay Register 18
"Seq0BDLY19_pX (for X = 0; X <= 3)" on page 990	0x90000+0x8f3+(X*0x100000)	Seq0BDLY19_pX: PHY Initialization Engine (PIE) Delay Register 19
"Seq0BDLY20_pX (for X = 0; X <= 3)" on page 991	0x90000+0x8f4+(X*0x100000)	Seq0BDLY20_pX: PHY Initialization Engine (PIE) Delay Register 20

Register	Offset	Description
"Seq0BDLY21_pX (for X = 0; X <= 3)" on page 992	0x90000+0x8f5+(X*0x100000)	Seq0BDLY21_pX: PHY Initialization Engine (PIE) Delay Register 21
"Seq0BDLY22_pX (for X = 0; X <= 3)" on page 993	0x90000+0x8f6+(X*0x100000)	Seq0BDLY22_pX: PHY Initialization Engine (PIE) Delay Register 22
"Seq0BDLY23_pX (for X = 0; X <= 3)" on page 994	0x90000+0x8f7+(X*0x100000)	Seq0BDLY23_pX: PHY Initialization Engine (PIE) Delay Register 23
"Seq0BDLY24_pX (for X = 0; X <= 3)" on page 995	0x90000+0x8f8+(X*0x100000)	Seq0BDLY24_pX: PHY Initialization Engine (PIE) Delay Register 24
"Seq0BDLY25_pX (for X = 0; X <= 3)" on page 996	0x90000+0x8f9+(X*0x100000)	Seq0BDLY25_pX: PHY Initialization Engine (PIE) Delay Register 25
"Seq0BDLY26_pX (for X = 0; X <= 3)" on page 997	0x90000+0x8fa+(X*0x100000)	Seq0BDLY26_pX: PHY Initialization Engine (PIE) Delay Register 26
"Seq0BDLY27_pX (for X = 0; X <= 3)" on page 998	0x90000+0x8fb+(X*0x100000)	Seq0BDLY27_pX: PHY Initialization Engine (PIE) Delay Register 27
"Seq0BDLY28_pX (for X = 0; X <= 3)" on page 999	0x90000+0x8fc+(X*0x100000)	Seq0BDLY28_pX: PHY Initialization Engine (PIE) Delay Register 28
"Seq0BDLY29_pX (for X = 0; X <= 3)" on page 1000	0x90000+0x8fd+(X*0x100000)	Seq0BDLY29_pX: PHY Initialization Engine (PIE) Delay Register 29
"Seq0BDLY30_pX (for X = 0; X <= 3)" on page 1001	0x90000+0x8fe+(X*0x100000)	Seq0BDLY30_pX: PHY Initialization Engine (PIE) Delay Register 30
"Seq0BDLY31_pX (for X = 0; X <= 3)" on page 1002	0x90000+0x8ff+(X*0x100000)	Seq0BDLY31_pX: PHY Initialization Engine (PIE) Delay Register 31
"UpdDisFlagMask" on page 1003	0x90000+0x900	UpdDisFlagMask: Programmed by dwc_ddrphy_phyinit_l_loadPIEImage() to support frequency...
"DisableFlagMask" on page 1004	0x90000+0x901	DisableFlagMask: Programmed by dwc_ddrphy_phyinit_l_loadPIEImage() to support frequency...

Register	Offset	Description
"DisableFlagMasksX (for X = 0; X <= 1)" on page 1005	0x90000+0x902	DisableFlagMasksX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency...
"RtrnMode_pX (for X = 0; X <= 3)" on page 1006	0x90000+0x903+(X*0x100000)	RtrnMode_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"RtrnModeMaskX (for X = 0; X <= 1)" on page 1007	0x90000+0x904+(X*0x82)	RtrnModeMaskX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"RtrnModeMask0sX (for X = 0; X <= 1)" on page 1008	0x90000+0x905	RtrnModeMask0sX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency...
"CtrlUpdAck" on page 1009	0x90000+0x912	CtrlUpdAck: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BResetFixedAddrBits_pX (for X = 0; X <= 3)" on page 1010	0x90000+0x938+(X*0x100000)	Seq0BResetFixedAddrBits_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency...
"Seq0BCntrX (for X = 0; X <= 3)" on page 1011	0x90000+0x950+(X*0x1)	Seq0BCntrX: Used by PIE during PPT2.
"Seq0BCntr0Threshold_pX (for X = 0; X <= 3)" on page 1012	0x90000+0x954+(X*0x100000)	Seq0BCntr0Threshold_pX: Used by PIE during PPT2.
"Seq0BCntr1Threshold_pX (for X = 0; X <= 3)" on page 1013	0x90000+0x955+(X*0x100000)	Seq0BCntr1Threshold_pX: Used by PIE during PPT2.
"Seq0BCntr2Threshold_pX (for X = 0; X <= 3)" on page 1014	0x90000+0x956+(X*0x100000)	Seq0BCntr2Threshold_pX: Used by PIE during PPT2.
"Seq0BCntr3Threshold_pX (for X = 0; X <= 3)" on page 1015	0x90000+0x957+(X*0x100000)	Seq0BCntr3Threshold_pX: Used by PIE during PPT2.
"Seq0BCntrCtrl" on page 1016	0x90000+0x958	Seq0BCntrCtrl: Used by PIE during PPT2.
"RtrnModeMask1sX (for X = 0; X <= 1)" on page 1017	0x90000+0x987	RtrnModeMask1sX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency...
"Seq0BGPR32_pX (for X = 0; X <= 3)" on page 1018	0x90000+0xa20+(X*0x100000)	Seq0BGPR32_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.

Register	Offset	Description
"Seq0BGPR33_pX (for X = 0; X <= 3)" on page 1019	0x90000+ 0xa21+(X* 0x100000)	Seq0BGPR33_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR34_pX (for X = 0; X <= 3)" on page 1020	0x90000+ 0xa22+(X* 0x100000)	Seq0BGPR34_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR35_pX (for X = 0; X <= 3)" on page 1021	0x90000+ 0xa23+(X* 0x100000)	Seq0BGPR35_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR36_pX (for X = 0; X <= 3)" on page 1022	0x90000+ 0xa24+(X* 0x100000)	Seq0BGPR36_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR37_pX (for X = 0; X <= 3)" on page 1023	0x90000+ 0xa25+(X* 0x100000)	Seq0BGPR37_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR38_pX (for X = 0; X <= 3)" on page 1024	0x90000+ 0xa26+(X* 0x100000)	Seq0BGPR38_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR39_pX (for X = 0; X <= 3)" on page 1025	0x90000+ 0xa27+(X* 0x100000)	Seq0BGPR39_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR40_pX (for X = 0; X <= 3)" on page 1026	0x90000+ 0xa28+(X* 0x100000)	Seq0BGPR40_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR41_pX (for X = 0; X <= 3)" on page 1027	0x90000+ 0xa29+(X* 0x100000)	Seq0BGPR41_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR42_pX (for X = 0; X <= 3)" on page 1028	0x90000+ 0xa2a+(X* 0x100000)	Seq0BGPR42_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR43_pX (for X = 0; X <= 3)" on page 1029	0x90000+ 0xa2b+(X* 0x100000)	Seq0BGPR43_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR44_pX (for X = 0; X <= 3)" on page 1030	0x90000+ 0xa2c+(X* 0x100000)	Seq0BGPR44_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR45_pX (for X = 0; X <= 3)" on page 1031	0x90000+ 0xa2d+(X* 0x100000)	Seq0BGPR45_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.

Register	Offset	Description
"Seq0BGPR46_pX (for X = 0; X <= 3)" on page 1032	0x90000+ 0xa2e+(X* 0x100000)	Seq0BGPR46_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"Seq0BGPR47_pX (for X = 0; X <= 3)" on page 1033	0x90000+ 0xa2f+(X* 0x100000)	Seq0BGPR47_pX: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
DWC_DDRPHYA_ZCAL0_p0 DesignWare Cores LPDDR5X/5/4X PHY address block DWC_DDRPHYA_ZCALj_Pk (for j == 0)(for k = 0; k <= 3) Exists: Always		
"ZcalClkDiv" on page 1035	0xb0000+ 0x1	ZcalClkDiv: Power-saving clock divider to ZCAL
"ZCalClkInfo_pX (for X = 0; X <= 3)" on page 1036	0xb0000+ 0x4+(X*0x 100000)	Impedance Calibration Clock Ratio
"MtestMuxSel" on page 1037	0xb0000+ 0x1a	MtestMuxSel: Digital Observation Pin control
"ZCALParityInvert" on page 1038	0xb0000+ 0x4d	ZCALParityInvert: Invert APB Parity for register slave ZCAL
"ScratchPadZCAL" on page 1039	0xb0000+ 0x7d	ScratchPadZCAL: ScratchPad for ZCAL
"ZCALReservedX (for X == 0)" on page 1040	0xb0000+ 0xfc	ZCALReservedX: Reserved for future use
"PUBReservedP1_pX (ZCAL) (for X = 0; X <= 3)" on page 1041	0xb0000+ 0xff+(X*0x 100000)	PUBReservedP1_pX: Reserved for future use
"ZCalBaseCtrl" on page 1042	0xb0000+ 0x301	Impedance Calibration control
"ZCalRate" on page 1044	0xb0000+ 0x303	Impedance Calibration timing control
"ZCalCtrl" on page 1046	0xb0000+ 0x305	Code Search related impedance calibration controls
"ZCalComplnvert" on page 1048	0xb0000+ 0x306	Impedance Calibration Comparator Invert control
"ZCalCompOffset" on page 1049	0xb0000+ 0x307	Impedance Calibration Comparator offset control
"ZCalPUOffset" on page 1050	0xb0000+ 0x308	Impedance Calibration Pull-up offset control

Register	Offset	Description
"ZCalCompOvr" on page 1051	0xb0000+0x309	ZCalCompOvr: Impedance Calibrator's comparator calibration override control
"ZCalPUOvr" on page 1052	0xb0000+0x30a	ZCalPUOvr: Impedance Calibrator's pull-up calibration override control
"ZCalClrFirstRunDone" on page 1053	0xb0000+0x30d	ZCalClrFirstRunDone: Clear FirstRunDone flag, inside Impedance Calibration control logic
"ZCalPDSearchGainTV" on page 1054	0xb0000+0x30e	ZCalPDSearchGainTV: Search code end-gain control for pull-down calibration
"ZCalAnaSettlingTime" on page 1056	0xb0000+0x30f	ZCalAnaSettlingTime: ZCalana settling time
"ZCalReset" on page 1057	0xb0000+0x310	ZCalReset: Impedance Calibration Off/Reset
"ZCalRun" on page 1058	0xb0000+0x311	ZCalRun: Impedance Calibration trigger
"ZCalBusy" on page 1059	0xb0000+0x312	ZCalBusy: Impedance Calibration Busy Status
"ZCalCompResult" on page 1060	0xb0000+0x313	ZCalCompResult: Impedance Calibrator comparator calibration code monitor
"ZCalPUResult" on page 1061	0xb0000+0x314	ZCalPUResult: Impedance Calibrator pull-up calibration code monitor
"ZCalPDResult" on page 1062	0xb0000+0x315	ZCalPDResult: Impedance Calibrator pull-down calibration code monitor
"ZCalCodePU" on page 1063	0xb0000+0x316	ZCalCodePU: Pull-up impedance calibration code from the calibration circuit
"ZCalCodePD" on page 1064	0xb0000+0x317	ZCalCodePD: Pull-down impedance calibration code from the calibration circuit
"ZCalCompSearchSeed" on page 1065	0xb0000+0x318	ZCalCompSearchSeed: Seed control for comparator offset calibration
"ZCalPUSearchSeed" on page 1066	0xb0000+0x319	ZCalPUSearchSeed: Seed control for pull-up calibration
"ZCalPDSearchSeed" on page 1067	0xb0000+0x31a	ZCalPDSearchSeed: Seed control for pull-down calibration
"ZCalCompSearchGainIV" on page 1068	0xb0000+0x31b	ZCalCompSearchGainIV: Search code initial-gain control for comparator offset calibration
"ZCalCompSearchGainTV" on page 1070	0xb0000+0x31c	ZCalCompSearchGainTV: Search code end-gain control for comparator offset calibration

Register	Offset	Description
"ZCalPUSearchGainIV" on page 1072	0xb0000+0x31d	ZCalPUSearchGainIV: Search code initial-gain control for pull-up calibration
"ZCalPUSearchGainTV" on page 1074	0xb0000+0x31e	ZCalPUSearchGainTV: Search code end-gain control for pull-up calibration
"ZCalPDSearchGainIV" on page 1076	0xb0000+0x31f	ZCalPDSearchGainIV: Search code initial-gain control for pull-down calibration
"ZQUpdate" on page 1078	0xb0000+0x320	ZQUpdate: Impedance Update
"ZQCalCodePUchX (for X = 0; X <= 1)" on page 1079	0xb0000+0x321+(X*0xc)	ZQCalCodePUchX: Per channel Pull-up impedance calibration code sent to all the IOs
"ZQCalCodePDchX (for X = 0; X <= 1)" on page 1080	0xb0000+0x322+(X*0xc)	ZQCalCodePDchX: channel 0 Pull-down impedance calibration code sent to all the IOs
"ZQCalBaseCtrl" on page 1081	0xb0000+0x323	IOs Calibration Base legs control
"ZQCalCodeOffsetPU" on page 1082	0xb0000+0x324	Pull-up impedance calibration code offset control
"ZQCalCodeOffsetPD" on page 1083	0xb0000+0x325	Pull-down impedance calibration code offset control
"ZQCalCodeOvrPU" on page 1084	0xb0000+0x326	Pull-up impedance Calibration code override
"ZQCalCodeOvrPD" on page 1085	0xb0000+0x327	Pull-down impedance Calibration code override
"ZQCalCodePUMax" on page 1086	0xb0000+0x328	ZQCalCodePUMax: Max ZQCalCodePU value for overvoltage protection
"ZQCalCodePUMin" on page 1087	0xb0000+0x329	ZQCalCodePUMin: Min ZQCalCodePU value for overvoltage protection
"ZQCalCodePDMax" on page 1088	0xb0000+0x32a	ZQCalCodePDMax: Max ZQCalCodePD value for overvoltage protection
"ZQCalCodePDMIN" on page 1089	0xb0000+0x32b	ZQCalCodePDMIN: Min ZQCalCodePD value for overvoltage protection
"ZCalStopClk_pX (for X = 0; X <= 3)" on page 1090	0xb0000+0x32f+(X*0x100000)	ZCalStopClk_pX: pstateable impedance calibration trigger to gate dficlk

Register	Offset	Description
		DWC_DDRPHYA_DRTUB\${inst} DesignWare Cores LPDDR5X/5/4X PHY address block DWC_DDRPHYA_DRTUBj (for j == 0) Exists: Always
<a href="#">"PieVecCfg" on page 1092</a>	0xc0000+ 0x0	Start vector value to be used for LP3-exit or Init PIE Sequence
<a href="#">"PieInitVecSel" on page 1093</a>	0xc0000+ 0x1	Start vector value to be used for LP3-exit or Init PIE Sequence
<a href="#">"DctShadowRegs" on page 1094</a>	0xc0000+ 0x4	DctShadowRegs: Reserved for PHY training firmware use.
<a href="#">"DctWriteOnlyShadow" on page 1095</a>	0xc0000+ 0x30	DctWriteOnlyShadow: Reserved for PHY training firmware use.
<a href="#">"UctWriteOnly" on page 1096</a>	0xc0000+ 0x32	UctWriteOnly: Reserved for PHY training firmware use.
<a href="#">"UctWriteProt" on page 1097</a>	0xc0000+ 0x33	UctWriteProt: Reserved for PHY training firmware use.
<a href="#">"UctDatWriteOnly" on page 1098</a>	0xc0000+ 0x34	UctDatWriteOnly: Reserved for PHY training firmware use.
<a href="#">"UctDatWriteProt" on page 1099</a>	0xc0000+ 0x35	UctDatWriteProt: Reserved for PHY training firmware use.
<a href="#">"UctlErr" on page 1100</a>	0xc0000+ 0x36	UctlErr: Reserved for PHY training firmware use.
<a href="#">"DRTUBParityInvert" on page 1101</a>	0xc0000+ 0x4d	DRTUBParityInvert: Invert APB Parity for register slave DRTUB
<a href="#">"UCParityInvert" on page 1102</a>	0xc0000+ 0x4e	UCParityInvert: Invert APB Parity for ARC *CCM Reads
<a href="#">"ScratchPadDRTUB" on page 1103</a>	0xc0000+ 0x7d	ScratchPadDRTUB: ScratchPad for DRTUB
<a href="#">"UcclkHclkEnables" on page 1104</a>	0xc0000+ 0x80	Ucclk and Hclk enables
<a href="#">"ArcEccIndications" on page 1105</a>	0xc0000+ 0x82	Indications of ARC double-bit ECC errors
<a href="#">"ArclccmSbErrCtr" on page 1106</a>	0xc0000+ 0x83	ArclccmSbErrCtr: Count of ICCM Single-bit corrected ECC Errors
<a href="#">"ArcDccmSbErrCtr" on page 1107</a>	0xc0000+ 0x84	ArcDccmSbErrCtr: Count of DCCM Single-bit corrected ECC Errors
<a href="#">"ArcSbCtrEnables" on page 1108</a>	0xc0000+ 0x85	Enables for single-bit error counters

Register	Offset	Description
"ArcPmuEccCtl" on page 1109	0xc0000+0x86	ArcPmuEccCtl: ARC ECC Control
"StartDCCMClear" on page 1110	0xc0000+0x88	StartDCCMClear: Enables the clearing of the DCCM by the PHY
"DCCMClearRunning" on page 1111	0xc0000+0x89	DCCMClearRunning: Indicates when DCCM clearing is in progress.
"PIEMicroReset" on page 1112	0xc0000+0x99	Controls reset and clock shutdown on the local microcontroller by the PIE.
"CUSTPHYREV" on page 1114	0xc0000+0xec	CUSTPHYREV: Customer settable by the customer
"CUSTPUBREV" on page 1115	0xc0000+0xed	CUSTPUBREV: Customer settable by the customer
"PUBREV" on page 1116	0xc0000+0xee	PUBREV: The hardware version of this PUB, excluding the PHY
"PUBVAR" on page 1117	0xc0000+0xef	PUBVAR: The hardware variant of this PUB
"DfiFreqXlatX (for X = 0; X <= 15)" on page 1118	0xc0000+0xf0+(X*0x1)	DFI Frequency Translation Register 0
DWC_DDRPHYA_APBONLY\${inst} DesignWare Cores LPDDR5X/5/4X PHY address block DWC_DDRPHYA_APBONLYj (for j == 0) Exists: Always		
"MicroContMuxSel" on page 1120	0xd0000+0x0	MicroContMuxSel: PMU Config Mux Select
"ContextToMicro" on page 1121	0xd0000+0x1	ContextToMicro: Determines which set of CSRs the ARC has access to
"ExternalAHBReset" on page 1122	0xd0000+0x2	ExternalAHBReset: Directly connected to Reset pin of external AHB-Lite interface
"TDRDisable" on page 1123	0xd0000+0x3	TDRDisable: Disables TDR Interface
"UctShadowRegs" on page 1124	0xd0000+0x4	PMU/Controller Protocol - Controller Read-only Shadow
"BlockDfiShadowRegs" on page 1125	0xd0000+0x5	BlockDfiInterface - Read-only Shadow
"CCMWriteBypassEnable" on page 1126	0xd0000+0x8	CCMWriteBypassEnable: ARC Write Bypass Enable

Register	Offset	Description
"DctWriteOnly" on page 1127	0xd0000+0x30	DctWriteOnly: Reserved for future use.
"DctWriteProt" on page 1128	0xd0000+0x31	DctWriteProt: DCT downstream mailbox protocol CSR.
"UctWriteOnlyShadow" on page 1129	0xd0000+0x32	UctWriteOnlyShadow: Read-only view of the csr UctDatWriteOnly
"UctDatWriteOnlyShadow" on page 1130	0xd0000+0x34	UctDatWriteOnlyShadow: Read-only view of the csr UctDatWriteOnly
"NeverGateCsrClock" on page 1131	0xd0000+0x35	NeverGateCsrClock: Reserved for PHY training firmware use.
"NeverGateAcCsrClock" on page 1132	0xd0000+0x36	NeverGateAcCsrClock: Reserved for PHY training firmware use.
"DfiCfgRdDataValidTicks" on page 1133	0xd0000+0x37	Bounding number of DfiClk ticks required for valid csr Rd Data on APB reads.
"DisableHMRdSpeedUp" on page 1134	0xd0000+0x39	DisableHMRdSpeedUp: Deprecated.
"OverrideHMRdSpeedUp" on page 1135	0xd0000+0x3a	OverrideHMRdSpeedUp: Override for logic that disables CSR Read Speedup for older Hard Macros
"Dfi0DebugControl" on page 1136	0xd0000+0x40	Dfi0DebugControl: Reserved for PHY training firmware use and for debugging
"Dfi0DebugCaptureX (for X = 0; X <= 1)" on page 1137	0xd0000+0x41+(X*0x1)	Dfi0DebugCaptureX: Reserved for PHY training firmware use and for debugging
"Dfi0DebugPerfCtrEn" on page 1138	0xd0000+0x43	Dfi0DebugPerfCtrEn: Reserved for PHY training firmware use and for debugging
"Dfi0DebugPerfCtr" on page 1139	0xd0000+0x44	Dfi0DebugPerfCtr: Reserved for PHY training firmware use and for debugging
"Dfi1DebugControl" on page 1140	0xd0000+0x48	Dfi1DebugControl: Reserved for PHY training firmware use and for debugging
"Dfi1DebugCaptureX (for X = 0; X <= 1)" on page 1141	0xd0000+0x49+(X*0x1)	Dfi1DebugCaptureX: Reserved for PHY training firmware use and for debugging
"Dfi1DebugPerfCtrEn" on page 1142	0xd0000+0x4b	Dfi1DebugPerfCtrEn: Reserved for PHY training firmware use and for debugging
"Dfi1DebugPerfCtr" on page 1143	0xd0000+0x4c	Dfi1DebugPerfCtr: Reserved for PHY training firmware use and for debugging
"APBONLYParityInvert" on page 1144	0xd0000+0x4d	APBONLYParityInvert: Invert APB Parity for register slave APBONLY

Register	Offset	Description
"WaitCondAPB" on page 1145	0xd0000+0x50	WaitCondAPB: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.
"PIERdDataValidTicks" on page 1146	0xd0000+0x51	Number of DfiClks between CfgBusRd and CfgBusDone on PIE reads
"UCRdDataValidTicks" on page 1147	0xd0000+0x52	Backup timeout counter for ARC register reads
"ScratchPadAPBONLY" on page 1148	0xd0000+0x7d	ScratchPadAPBONLY: ScratchPad for APBONLY
"MicroReset" on page 1149	0xd0000+0x99	Controls reset and clock shutdown on the local microcontroller
"MicroResetPIEEn" on page 1151	0xd0000+0x9a	MicroResetPIEEn: Enables use PIE-specific ARC controls
"ClearPIESTallToMicro" on page 1152	0xd0000+0x9b	ClearPIESTallToMicro: Forces csrPIESTallToMicro to 0
"PIEMicroStallDelay" on page 1153	0xd0000+0x9c	PIEMicroStallDelay: Delay before PIESTallToMicro takes effect
"SequencerOverride" on page 1154	0xd0000+0xe7	SequencerOverride: Reserved for PHY training firmware use.
"DfiInitCompleteShadow" on page 1155	0xd0000+0xfa	DfiInitCompleteShadow: dfi_init_complete - Controller Read-only Shadow
"APBONLYReservedX (for X == 0)" on page 1156	0xd0000+0xfd	APBONLYReservedX: Reserved for future use

## 13.1 **DWC\_DDRPHYA\_DBYTEj\_Pk Registers**

### 13.1.1 DFIMRL\_pX (for X = 0; X <= 3)

- **Description:** DFIMRL\_pX: DFI MaxReadLatency
- **Size:** 6 bits
- **Offset:** (0x10000+(j<<12))+0x0+(X\*0x100000)
- **Exists:** Always



**Table 13-5 Fields for Register: DFIMRL\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
5:0	DFIMRL_p0	R/W	<p>DFIMRL_p0: This Max Read Latency CSR is to be trained to ensure the rx-data FIFO is not read until after all dbytes have their read data valid. In DBYTE, this is the value, in units of DFI Clocks (DfiClk), between dfi_rddata_en and dfi_rddata_valid. Maximum supported value is 35 decimal.</p> <p><b>Value After Reset:</b> 0x6</p> <p><b>Exists:</b> Always</p>

### 13.1.2 EnableWriteLinkEcc\_pX (for X = 0; X <= 3)

- **Description:** EnableWriteLinkEcc\_pX: Enable Write Data Link ECC. Applicable only in LPDDR5 Protocol
- **Size:** 1 bit
- **Offset:**  $(0x10000 + (j \ll 12)) + 0x1 + (X * 0x100000)$
- **Exists:** Always



**Table 13-6 Fields for Register: EnableWriteLinkEcc\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
0	EnableWriteLinkEcc_p0	R/W	<p>EnableWriteLinkEcc_p0: When 1, dfi_wrdata_link_ecc input will be transmitted on RDQST. The LPDDR5 DRAM MR22.WECC needs to be configured accordingly.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.3 DbyteMiscMode

- **Name:** DBYTE Module Disable
- **Description:** DbyteMiscMode: Controls the enable/disable state of the DBYTE module.
- **Size:** 3 bits
- **Offset:** (0x10000+(j<<12))+0x2
- **Exists:** Always

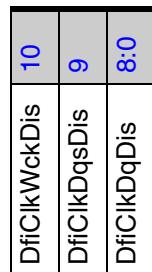


**Table 13-7 Fields for Register: DbyteMiscMode**

Bits	Name	Memory Access	Description
2	DByteDisable	R/W	<p>DByteDisable: Controls whether this DBYTE module is disabled. If this DBYTE module is not enabled, it receives no clocks and remains in reset.</p> <ul style="list-style-type: none"> <li>■ 0 - Enable this DBYTE module</li> <li>■ 1 - Disable this DBYTE module</li> </ul> <p>This field should only be changed PHY initialization step C.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
1:0	Reserved	R	<p>Reserved: Reserved</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.4 DxDfiClkDis\_pX (for X = 0; X <= 3)

- **Name:** DBYTE DfiClk clock Disable
- **Description:** Dx DfiClkDis\_pX: Controls the enable/disable state of the DfiClk in DBYTE module. This field should only be changed PHY initialization step C.
- **Size:** 11 bits
- **Offset:** (0x10000+(j<<12))+0x3+(X\*0x100000)
- **Exists:** Always

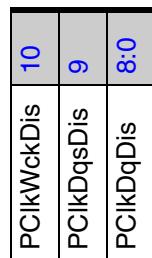


**Table 13-8 Fields for Register: Dx DfiClkDis\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
10	DfiClkWckDis	R/W	<p>DfiClkWckDis: Signal which controls WCK DIFF DfiClk.  <b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
9	DfiClkDqsDis	R/W	<p>DfiClkDqsDis: Signal which controls DQS DIFF DfiClk.  <b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
8:0	DfiClkDqDis	R/W	<p>DfiClkDqDis: Signal which controls SE DfiClk.</p> <ul style="list-style-type: none"> <li>■ 7:0 -&gt; dq[7:0]</li> <li>■ 8 -&gt; dbi</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.1.5 DxPClkDis\_pX (for X = 0; X <= 3)

- **Name:** DBYTE PClk clock Disable
- **Description:** DxPClkDis\_pX: Controls the enable/disable state of the PClk in DBYTE module. This field should only be changed PHY initialization step C. Corresponding DfiClkDis bit needs to be 0 when the toggling bit in this CSR.
- **Size:** 11 bits
- **Offset:** (0x10000+(j<<12))+0x4+(X\*0x100000)
- **Exists:** Always



**Table 13-9 Fields for Register: DxPClkDis\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
10	PClkWckDis	R/W	PClkWckDis: Signal which controls WCK DIFF PClk <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
9	PClkDqsDis	R/W	PClkDqsDis: Signal which controls DQS DIFF PClk. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
8:0	PClkDqDis	R/W	PClkDqDis: Signal which controls SE PClk. <ul style="list-style-type: none"> <li>■ 7:0 -&gt; dq[7:0]</li> <li>■ 8 -&gt; dbi</li> </ul> <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.6 LP5DfiDataEnLatency\_pX (for X = 0; X <= 3)

- **Name:** Selects the depth of the data pipes in the PUB relative to AC pipes
- **Description:** LP5DfiDataEnLatency\_pX: The read data pipe depth must be consistent with DRAM read latency RL and the relative assertions of dfi\_cs and dfi\_rddata\_en. The write data pipe depth must be consistent with DRAM write latency WL and the relative assertions of dfi\_cs and dfi\_wrdata\_en. The pipeline stages are required for generation or reception of preambles, and for changing LCDL timing, and for the power saving feature of RxStandby (csrDxRxStandbyEn).
- **Size:** 1 bit
- **Offset:**  $(0x10000 + (j \ll 12)) + 0x8 + (X * 0x100000)$
- **Exists:** Always



**Table 13-10 Fields for Register: LP5DfiDataEnLatency\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
0	LP5RLm13	R/W	<p>LP5RLm13: When asserted the LP5 data read pipe will be 13 tWCK longer than the AC pipe, else the LP5 data read pipe will be 5 tWCK longer than the AC pipe.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.7 DfiCtrlRx\_fifoRst

- **Description:** DfiCtrlRx\_fifoRst: Reset RxFifos on sideband transaction for LPDDR54
- **Size:** 1 bit
- **Offset:** (0x10000+(j<<12))+0xb
- **Exists:** Always



**Table 13-11 Fields for Register: DfiCtrlRx\_fifoRst**

Bits	Name	Memory Access	Description
0	DfiCtrlRx_fifoRst	R/W	<p>DfiCtrlRx_fifoRst: When set, a phyupd or ctrlupd resets the read and write pointers of the dbyte read-data FIFO and of the read credit FIFO, which is the FIFO of the dqsen_sync that extends the read rxen. These state variables may also be reset by csrRxFifoInit</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.8 PptDqsCntInvTrnTg0\_pX (for X = 0; X <= 3)

- **Description:** PptDqsCntInvTrnTg0\_pX: Programmed by PHY training firmware to support LPDDR4X/5 DRAM drift compensation.
- **Size:** 16 bits
- **Offset:**  $(0x10000 + (j \ll 12)) + 0xc + (X * 0x100000)$
- **Exists:** Always



**Table 13-12 Fields for Register: PptDqsCntInvTrnTg0\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	PptDqsCntInvTrnTg0_p0	R/W	<p>PptDqsCntInvTrnTg0_p0: Programmed by PHY training firmware to support LPDDR4X/5 DRAM drift compensation.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.9 PptDqsCntInvTrnTg1\_pX (for X = 0; X <= 3)

- **Description:** PptDqsCntInvTrnTg1\_pX: Programmed by PHY training firmware to support LPDDR4X/5 DRAM drift compensation.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xd+(X\*0x100000)
- **Exists:** Always



**Table 13-13 Fields for Register: PptDqsCntInvTrnTg1\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	PptDqsCntInvTrnTg1_p0	R/W	<p>PptDqsCntInvTrnTg1_p0: Programmed by PHY training firmware to support LPDDR4X/5 DRAM drift compensation.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.10 TrackingModeCntrl\_pX (for X = 0; X <= 3)

- **Name:** Mode controls for fine RxEn timing adjustment during mission-mode operation.
- **Description:** TrackingModeCntrl\_pX: Mode controls for fine RxEn timing adjustment during mission-mode operation. For tracking variations in the DRAM tDQSCK.
- **Size:** 15 bits
- **Offset:**  $(0x10000 + (j \ll 12)) + 0xe + (X * 0x100000)$
- **Exists:** Always

RxDqsTrackingThreshold	14:12
DqsOscRunTimeSel	11:8
Tdq2dqsTrackingLimit	7:5
ReservedTrackingModeCntr	4
Twck2dqo TrackingLimit	3:1
EnWck2DqoSnoopTracking	0

Table 13-14 Fields for Register: TrackingModeCntrl\_pX (for X = 0; X <= 3)

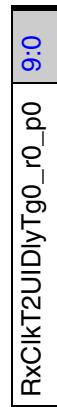
Bits	Name	Memory Access	Description
14:12	RxDqsTrackingThreshold	R/W	<p>RxDqsTrackingThreshold: When EnRxDqsTracking=1, this csr is used to set the max value of the signed count of the sense of the count.</p> <ul style="list-style-type: none"> <li>■ 0 3</li> <li>■ 1 5, default value</li> <li>■ 2 9</li> <li>■ 3 15</li> <li>■ 4 31</li> <li>■ 5-7 reserved</li> </ul> <p><b>Value After Reset:</b> 0x1  <b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
11:8	DqsOscRunTimeSel	R/W	<p>DqsOscRunTimeSel: Informs the PHY of the DQS oscillator runtime configured in the DRAMs, for correct calculations using the LPDDR5 MR35/36 and MR38/39; See JEDEC 7.6.14. See JEDEC LPDDR5 MR37/40 for WCK2DQI/WCK2DQO Interval timer run time. See JEDEC 7.6.14 specifies that the WCK2DQx Interval Oscillator counts the number of CK cycles. PHY supports only DqsOscRunTimeSel = 0x3 for both initial training and retraining.</p> <ul style="list-style-type: none"> <li>■ 0 for a runtime of 256 memCK (Reserved)</li> <li>■ 1 for a runtime of 512 memCK (Reserved)</li> <li>■ 2 for a runtime of 1024 memCK (Reserved)</li> <li>■ 3 for a runtime of 2048 memCK (default)</li> <li>■ 4 for a runtime of 4096 memCK (Reserved)</li> <li>■ 5 for a runtime of 8192 memCK (Reserved)</li> <li>■ 6..15 reserved</li> </ul> <p><b>Value After Reset:</b> 0x3</p> <p><b>Exists:</b> Always</p>
7:5	Tdq2dqTrackingLimit	R/W	<p>Tdq2dqTrackingLimit: When tracking the dram tdqs2dq in LP4X or twck2dqi in LP5, in either dfi-controlled mode or phy-controlled mode, limit the change to the Tx DQ delay that may be applied at each update. This does not limit the total excursion from the trained value of TxDqDly.</p> <ul style="list-style-type: none"> <li>■ 0 no limit (default)</li> <li>■ 1 1/64 UI</li> <li>■ 2 2/64 UI</li> <li>■ 3 4/64 UI</li> <li>■ 4 8/64 UI</li> <li>■ 5 16/64 UI</li> <li>■ 6,7 reserved</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
4	ReservedTrackingModeCntrl	R/W	<p>ReservedTrackingModeCntrl: Reserved for future use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
3:1	Twck2dqoTrackingLimit	R/W	<p>Twck2dqoTrackingLimit: When tracking the dram twck2dqo in LP5, in either dfi-controlled mode or phy-controlled mode, limit the change to the RxClk delay that may be applied at each update. This does not limit the total excursion from the trained value of TxDqDly.</p> <ul style="list-style-type: none"> <li>■ 0 no limit (default)</li> <li>■ 1 1/64 UI</li> <li>■ 2 2/64 UI</li> <li>■ 3 4/64 UI</li> <li>■ 4 8/64 UI</li> <li>■ 5 16/64 UI</li> <li>■ 6,7 reserved</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
0	EnWck2DqoSnoopTracking	R/W	<p>EnWck2DqoSnoopTracking: Enable Snoop Interface to modify the RxEn timing</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.1.11 RxClkT2UIDlyTg0\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)

- **Description:** RxClkT2UIDlyTg0\_rY\_pX: Trained Read DQS to RxClk Delay (Timing Group DEST=0).
- **Size:** 10 bits
- **Offset:** (0x10000+(j<<12))+0x10+(X\*0x100000)+(Y\*0x100)
- **Exists:** Always

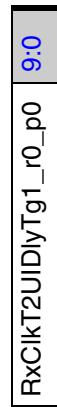


**Table 13-15 Fields for Register: RxClkT2UIDlyTg0\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)**

Bits	Name	Memory Access	Description
9:0	RxClkT2UIDlyTg0_r0_p0	R/W	<p>RxClkT2UIDlyTg0_r0_p0: Trained Read DQS_t to RxClk_t Delay (Timing Group DEST=0). Trained to generate read DQ receive clock from the received read DQS.</p> <ul style="list-style-type: none"> <li>■ RxClkDlyTg0[6:0] is the fine delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> </ul> <p>Fine delay may range from 0 to 2UI-noninclusive.  RxClkDlyTg0[9:7] is the coarse delay, i.e., one unit of delay is 1 UI. There are two values of this csr that result in the same delay.  For example 001_0000000 and 000_1000000 both result in an RxClk at the same time (nominally).  For example 010_0000000 and 001_1000000 both result in an RxClk at the same time (nominally). The training firmware is responsible for changing the coarse field [9:7] as required.  These Registers are replicated per pstate, per timing group, per dq lane.</p> <p><b>Note:</b> Timing Group 0 = RANK0</p> <ul style="list-style-type: none"> <li>■ RxClkT2UIDlyTg0[9:7]=0 use the first posedge DQS_t after RxEn to generate the RxClk for first datum</li> <li>■ RxClkT2UIDlyTg0[9:7]=1 use the first posedge DQS_c after RxEn to generate the RxClk for first datum</li> <li>■ RxClkT2UIDlyTg0[9:7]=2 use the second posedge DQS_t after RxEn to generate the RxClk for first datum</li> <li>■ RxClkT2UIDlyTg0[9:7]=3 use the second posedge DQS_c after RxEn to generate the RxClk for first datum</li> <li>■ RxClkT2UIDlyTg0[9:7]=4 use the third posedge DQS_t after RxEn to generate the RxClk for first datum</li> </ul> <p><b>Value After Reset:</b> 0x110  <b>Exists:</b> Always</p>

### 13.1.12 RxClkT2UIDlyTg1\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)

- **Description:** RxClkT2UIDlyTg1\_rY\_pX: Trained Read DQS to RxClk Delay (Timing Group DEST=1).
- **Size:** 10 bits
- **Offset:** (0x10000+(j<<12))+0x11+(X\*0x100000)+(Y\*0x100)
- **Exists:** Always

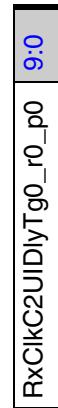


**Table 13-16 Fields for Register: RxClkT2UIDlyTg1\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)**

Bits	Name	Memory Access	Description
9:0	RxClkT2UIDlyTg1_r0_p0	R/W	<p>RxClkT2UIDlyTg1_r0_p0: Trained Read DQS_t to RxClk_t Delay (Timing Group DEST=1). Trained to generate read DQ receive clock from the received read DQS.</p> <ul style="list-style-type: none"> <li>■ RxClkDlyTg1[6:0] is the fine delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> </ul> <p>Fine delay may range from 0 to 2UI-noninclusive.  RxClkDlyTg1[9:7] is the coarse delay, i.e., one unit of delay is 1 UI. There are two values of this csr that result in the same delay.  For example 001_0000000 and 000_1000000 both result in an RxClk at the same time (nominally).  For example 010_0000000 and 001_1000000 both result in an RxClk at the same time (nominally). The training firmware is responsible for changing the coarse field [9:7] as required.  These Registers are replicated per pstate, per timing group, per dq lane.</p> <p><b>Note:</b> Timing Group 1 = RANK1</p> <ul style="list-style-type: none"> <li>■ RxClkT2UIDlyTg1[9:7]=0 use the first posedge DQS_t after RxEn to generate the RxClk for first datum</li> <li>■ RxClkT2UIDlyTg1[9:7]=1 use the first posedge DQS_c after RxEn to generate the RxClk for first datum</li> <li>■ RxClkT2UIDlyTg1[9:7]=2 use the second posedge DQS_t after RxEn to generate the RxClk for first datum</li> <li>■ RxClkT2UIDlyTg1[9:7]=3 use the second posedge DQS_c after RxEn to generate the RxClk for first datum</li> <li>■ RxClkT2UIDlyTg1[9:7]=4 use the third posedge DQS_t after RxEn to generate the RxClk for first datum</li> </ul> <p><b>Value After Reset:</b> 0x110  <b>Exists:</b> Always</p>

### 13.1.13 RxClkC2UIDlyTg0\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)

- **Description:** RxClkC2UIDlyTg0\_rY\_pX: Trained Read DQS\_c to RxClkc Delay (Timing Group DEST=0).
- **Size:** 10 bits
- **Offset:** (0x10000+(j<<12))+0x12+(X\*0x100000)+(Y\*0x100)
- **Exists:** Always

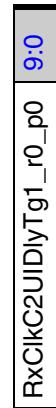


**Table 13-17 Fields for Register: RxClkC2UIDlyTg0\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)**

Bits	Name	Memory Access	Description
9:0	RxClkC2UIDlyTg0_r0_p0	R/W	<p>RxClkC2UIDlyTg0_r0_p0: Trained Read DQS_c to RxClkc Delay (Timing Group DEST=0). Trained to generate read DQ receive clock from the received read DQS_c.</p> <ul style="list-style-type: none"> <li>■ RxClkcDlyTg0[6:0] is the fine delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> </ul> <p>Fine delay may range from 0 to 2UI-noninclusive. These Registers are replicated per pstate, per timing group, per dq lane.</p> <p><b>Note:</b> Timing Group 0 = RANK0</p> <ul style="list-style-type: none"> <li>■ RxClkC2UIDlyTg0[9:7] are reserved.</li> </ul> <p><b>Value After Reset:</b> 0x110</p> <p><b>Exists:</b> Always</p>

### 13.1.14 RxClkC2UIDlyTg1\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)

- **Description:** RxClkC2UIDlyTg1\_rY\_pX: Trained Read DQS\_c to RxClkc Delay (Timing Group DEST=1).
- **Size:** 10 bits
- **Offset:** (0x10000+(j<<12))+0x13+(X\*0x100000)+(Y\*0x100)
- **Exists:** Always



**Table 13-18 Fields for Register: RxClkC2UIDlyTg1\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)**

Bits	Name	Memory Access	Description
9:0	RxClkC2UIDlyTg1_r0_p0	R/W	<p>RxClkC2UIDlyTg1_r0_p0: Trained Read DQS_c to RxClkc Delay (Timing Group DEST=1). Trained to generate read DQ receive clock from the received read DQS_c.</p> <ul style="list-style-type: none"> <li>■ RxClkcDlyTg1[6:0] is the fine delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> </ul> <p>Fine delay may range from 0 to 2UI-noninclusive. These Registers are replicated per pstate, per timing group, per dq lane.</p> <p><b>Note:</b> Timing Group 1 = RANK1</p> <ul style="list-style-type: none"> <li>■ RxClkC2UIDlyTg1[9:7] are reserved.</li> </ul> <p><b>Value After Reset:</b> 0x110</p> <p><b>Exists:</b> Always</p>

### 13.1.15 PptWck2DqoCntInvTrnTg0\_pX (for X = 0; X <= 3)

- **Description:** PptWck2DqoCntInvTrnTg0\_pX: Programmed by PHY training firmware to support LPDDR5 DRAM drift compensation.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0x14+(X\*0x100000)
- **Exists:** Always



**Table 13-19 Fields for Register: PptWck2DqoCntInvTrnTg0\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	PptWck2DqoCntInvTrnTg0_p0	R/W	PptWck2DqoCntInvTrnTg0_p0: Programmed by PHY training firmware to support LPDDR5 DRAM drift compensation. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.16 PptWck2DqoCntInvTrnTg1\_pX (for X = 0; X <= 3)

- **Description:** PptWck2DqoCntInvTrnTg1\_pX: Programmed by PHY training firmware to support LPDDR5 DRAM drift compensation.
- **Size:** 16 bits
- **Offset:**  $(0x10000 + (j \ll 12)) + 0x15 + (X * 0x100000)$
- **Exists:** Always



**Table 13-20 Fields for Register: PptWck2DqoCntInvTrnTg1\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	PptWck2DqoCntInvTrnTg1_p0	R/W	<p>PptWck2DqoCntInvTrnTg1_p0: Programmed by PHY training firmware to support LPDDR5 DRAM drift compensation.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.17 InitSeqControl

- **Name:** AC/DBYTE Pipeline Init Sequence/Pipeline Control register
- **Description:** InitSeqControl: This register directly controls whether the overall AC/DBYTE pipeline initialization sequence steps can be automatically triggered by TxRdPtrInit pipeline, or only controlled/set by PState sequencer by writing manually to each independent init control CSR below. The PState sequencer must write to each independent init control, if the TxPdPtrInit Pipeline is disabled. The PState sequencer must satisfy all the rules & restriction of the init control sequence as per describe in Arch spec - Ex. TxRdPtrInit=1 --> TxRdPtrInit=0 --> CMDFIFO running --> other init control. This is to allow the PState sequencer to have full controllability of the init sequence if needed.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0x16
- **Exists:** Always

ReservedInitSeqControl	15
InitControlRxReplSeqInit	14
InhibitTxRdPtrRxRepSeqInit	13
InitControlDbRxEnPhUpdInit	12
InitControlDbPptInit	11
InhibitTxRdPtrDbPptInit	10
InitControlDbDataPipeInit	9
InhibitTxRdPtrDbRxEnPhUpdInit	8
InhibitTxRdPtrDbDataPipeInit	7
InitControlTxXFIFOInit	6
InhibitTxRdPtrTXXFIFOInit	5
InitControlRxReplLcdlInit	4
InhibitTxRdPtrRxRepLcdlInit	3
InitControlRstLclCal	2
InhibitTxRdPtrRstLclCal	1
InhibitTxRdPtrBypassForce	0

Table 13-21 Fields for Register: InitSeqControl

Bits	Name	Memory Access	Description
15	ReservedInitSeqControl	R/W	ReservedInitSeqControl: Reserved Bits <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
14	InitControlRxReplSeqInit	R/W	InitControlRxReplSeqInit: When set (1), this sets the RxReplica calibration sequencer in Init State. This is not used by PUB AC block. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

Bits	Name	Memory Access	Description
13	InhibitTxRdPtrRxRepSeqInit	R/W	<p>InhibitTxRdPtrRxRepSeqInit: Inhibit the TxRdPtrInit Pipeline from resetting the RxReplica Sequencer When set (1), the PState sequencer may reset the LCDL calibration registers via InitControlRxRepSeqInit, in absence of the Pipeline Control This is not used by PUB AC block.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
12	InitControlDbRxEnPhUpdInit	R/W	<p>InitControlDbRxEnPhUpdInit: Whenever toggle (0-&gt;1), this creates a RxEnPhaseUpdate pulse to reset RxEnPhaseA/B logic Note that this does not changes/reset any RxEn Phase Delay selection/value This is not used by PUB AC block.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
11	InitControlDbPptInit	R/W	<p>InitControlDbPptInit: When set (1), this resets the DByte PPT block This is not used by PUB AC block.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
10	InhibitTxRdPtrDbPptInit	R/W	<p>InhibitTxRdPtrDbPptInit: Inhibit the TxRdPtrInit Pipeline from resetting the DByte PPT block. When set (1), The PState sequencer may reset the DByte PPT block via csrInitControlDbPptInit, in absence of the Pipeline Control This is not used by PUB AC block.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
9	InitControlDbDataPipeInit	R/W	<p>InitControlDbDataPipeInit: When set (1), this resets the internal data-pipe registers This is not used by PUB AC block.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
8	InhibitTxRdPtrDbRxEnPhUpdInit	R/W	<p>InhibitTxRdPtrDbRxEnPhUpdInit: Inhibit the TxRdPtrInit Pipeline from creating a RxEnPhaseUpdate pulse to reset RxEnPhaseA/B logic Note that this does not changes/reset any RxEn Phase Delay selection/value When set (1), The PState sequencer may reset the internal data-pipe registers via csrInitControlDbRxEnPhUpdInit, in absence of the Pipeline Control This is not used by PUB AC block.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
7	InhibitTxRdPtrDbDataPipeInit	R/W	<p>InhibitTxRdPtrDbDataPipeInit: Inhibit the TxRdPtrInit Pipeline from resetting the internal data-pipe registers When set (1), The PState sequencer may reset the internal data-pipe registers via csrInitControlDbDataPipeInit, in absence of the Pipeline Control This is not used by PUB AC block.</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
6	InitControlTXFIFOInit	R/W	<p>InitControlTXFIFOInit: Whenever toggle (0-&gt;1), this trigger 1 round of TXFIFO (DATA FIFO) content initialization cycle This min high-low pulse width of this InitControlTXFIFOInit must be min 26 DfiPubClk cycles</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
5	InhibitTxRdPtrTXFIFOInit	R/W	<p>InhibitTxRdPtrTXFIFOInit: Inhibit the TxRdPtrInit Pipeline from initializing the TXFIFO (DATA FIFO) content initialization When set (1), the PState sequencer may initialize the RxReplica Sequencer via csrInitControlTXFIFOInit, in absence of the Pipeline Control</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
4	InitControlRxReplLcdlInit	R/W	<p>InitControlRxReplLcdlInit: When set (1), this resets the RxReplica LCDL calibration registers (Pclk domain) This is not used by PUB AC block.</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
3	InhibitTxRdPtrRxReplLcdlInit	R/W	<p>InhibitTxRdPtrRxReplLcdlInit: Inhibit the TxRdPtrInit Pipeline from resetting the RxReplica LCDL calibration registers (Pclk domain) When set (1), the PState sequencer may reset the LCDL calibration registers via InitControlRxReplLcdlInit, in absence of the Pipeline Control This is not used by PUB AC block.</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
2	InitControlRstLclCal	R/W	<p>InitControlRstLclCal: When set (1), this resets the LCDL calibration registers (Pclk domain)</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
1	InhibitTxRdPtrRstLclCal	R/W	<p>InhibitTxRdPtrRstLclCal: Inhibit the TxRdPtrInit Pipeline from resetting the LCDL calibration registers (Pclk domain). When set (1), the PState sequencer may reset the LCDL calibration registers via csrlInitControlRstLclCal, in absence of the Pipeline Control</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
0	InhibitTxRdPtrBypassForce	R/W	<p>InhibitTxRdPtrBypassForce: Inhibit the TxRdPtrInit Pipeline from forcing the IO into Bypass Mode. When set (1), the PState sequencer may control the IO Bypass via csrAsync*TxMode, in absence of the Pipeline Control</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.18 TxDqsLeftEyeOffsetTg0\_pX (for X = 0; X <= 3)

- **Description:** TxDqsLeftEyeOffsetTg0\_pX: Write DQS Left Eye Offset (Timing Group 0).
- **Size:** 6 bits
- **Offset:** (0x10000+(j<<12))+0x19+(X\*0x100000)
- **Exists:** Always



**Table 13-22 Fields for Register: TxDqsLeftEyeOffsetTg0\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
5:0	TxDqsLeftEyeOffsetTg0_p0	R/W	<p>TxDqsLeftEyeOffsetTg0_p0: Write DQS Offset (Timing Group 0). Applicable in ECC Mode for LPDDR5. Program with a difference of the Mid-Point value and the Left Edge.</p> <ul style="list-style-type: none"> <li>■ Tx DQs Offset ( 6b fine delays). (For Timing Group 0)</li> <li>■ TxDqsLeftEySeOffsetTg0[5:0] is the fine (fractional UI) delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> </ul> <p><b>Note:</b> These Registers are replicated per pstate.  <b>Note:</b> Timing Group 0 = RANK0  <b>Value After Reset:</b> 0x1f  <b>Exists:</b> Always</p>

### 13.1.19 MtestMuxSel

- **Description:** MtestMuxSel: Digital Observation Pin control
- **Size:** 10 bits
- **Offset:** (0x10000+(j<<12))+0x1a
- **Exists:** Always



**Table 13-23 Fields for Register: MtestMuxSel**

Bits	Name	Memory Access	Description
9:0	MtestMuxSel	R/W	<p>MtestMuxSel: Controls for the mux for asynchronous data to the Digital Observation Pin.</p> <ul style="list-style-type: none"> <li>■ Encoding 9'h0 causes this chiplet to drive 0, (allowing flat 'OR' of pass-through information).</li> <li>■ MtestMuxSel[4:0] - Lower 5 bits selects one bit from the 32 bit MtestMux in each section or slice.</li> </ul> <p>Detailed tables are in the PUB Databook for PUB sections (AC, MASTER, etc.)</p> <ul style="list-style-type: none"> <li>■ MtestMuxSel[8:5] - Where more than one MtestMux exists, non-zero values select the outputs of the additional Mtest-Muxes,           <ul style="list-style-type: none"> <li>□ DBYTE MtestMuxSel[6:5]=2'h0 for Mux-A and 2'h1 for Mux-B and 2'h2 for Mux-C.</li> <li>□ For all other slaves, MtestMuxSel[8:5] are unused</li> </ul> </li> </ul> <p><b>Note:</b> See the PUB Databook for how, or if, the Digital Observation Pin is mapped to a physical bump in this configuration.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.20 TxDqsLeftEyeOffsetTg1\_pX (for X = 0; X <= 3)

- **Description:** TxDqsLeftEyeOffsetTg1\_pX: Write DQS Left Eye Offset (Timing Group 1).
- **Size:** 6 bits
- **Offset:** (0x10000+(j<<12))+0x1b+(X\*0x100000)
- **Exists:** Always



**Table 13-24 Fields for Register: TxDqsLeftEyeOffsetTg1\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
5:0	TxDqsLeftEyeOffsetTg1_p0	R/W	<p>TxDqsLeftEyeOffsetTg1_p0: Write DQS Offset (Timing Group 1). Applicable in ECC Mode for LPDDR5. Program with a difference of the Mid-Point value and the Left Edge.</p> <ul style="list-style-type: none"> <li>■ Tx DQS Offset ( 6b fine delays). (For Timing Group 1)</li> <li>■ TxDqsLeftEySeOffsetTg1[5:0] is the fine (fractional UI) delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> </ul> <p><b>Note:</b> These Registers are replicated per pstate.  <b>Note:</b> Timing Group 1 = RANK1  <b>Value After Reset:</b> 0x1f  <b>Exists:</b> Always</p>

### 13.1.21 RxEnDlyTg0\_pX (for X = 0; X <= 3)

- **Description:** RxEnDlyTg0\_pX: Trained Receive Enable Delay (For Timing Group 0)
- **Size:** 12 bits
- **Offset:** (0x10000+(j<<12))+0x20+(X\*0x100000)
- **Exists:** Always



**Table 13-25 Fields for Register: RxEnDlyTg0\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
11:0	RxEnDlyTg0_p0	R/W	<p>RxEnDlyTg0_p0: Trained Receive Enable Delay (For Timing Group 0) Trained to set the delay from the memory-read command to the signal enabling the read DQS to generate read-data strobes. The trained read DQS enable asserts during the read DQS preamble and deasserts at the end of the read burst.</p> <ul style="list-style-type: none"> <li>■ RxEnDlyTg0[11:7] is the coarse delay, i.e., one unit of delay is 2 UI. Max value of RxEnDlyTg0[11:7] = 31;</li> <li>■ RxEnDlyTg0[6:0] is the fine delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> <li>■ RxEnDlyTg0[6:0] delay range is 0 to minimum of 127/64 UI and max LCDL delay.</li> <li>■ RxEnDlyTg0[11:7] is also used when csr EnStrbLssRd-Mode=1 for coarse timing of the HM RxTimingSel.</li> <li>■ In that mode the RxEnDlyTg0 should be configured with the minimum across all lanes of the dbyte of the RxDigStrbDlyTg0.</li> </ul> <p><b>Note:</b> These Registers are replicated per pstate.  <b>Note:</b> Timing Group 0 = RANK0  <b>Value After Reset:</b> 0x100  <b>Exists:</b> Always</p>

### 13.1.22 RxEnDlyTg1\_pX (for X = 0; X <= 3)

- **Description:** RxEnDlyTg1\_pX: Trained Receive Enable Delay (For Timing Group 1)
- **Size:** 12 bits
- **Offset:** (0x10000+(j<<12))+0x21+(X\*0x100000)
- **Exists:** Always



**Table 13-26 Fields for Register: RxEnDlyTg1\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
11:0	RxEnDlyTg1_p0	R/W	<p>RxEnDlyTg1_p0: Trained Receive Enable Delay (For Timing Group 1) Trained to set the delay from the memory-read command to the signal enabling the read DQS to generate read-data strobes. The trained read DQS enable asserts during the read DQS preamble and deasserts at the end of the read burst.</p> <ul style="list-style-type: none"> <li>■ RxEnDlyTg1[11:7] is the coarse delay, i.e., one unit of delay is 2 UI. Max value of RxEnDlyTg1[11:7] = 31;</li> <li>■ RxEnDlyTg1[6:0] is the fine delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> <li>■ RxEnDlyTg1[6:0] delay range is 0 to minimum of 127/64 UI and max LCDL delay.</li> <li>■ RxEnDlyTg1[11:7] is also used when csr EnStrblssRd-Mode=1 for coarse timing of the HM RxTimingSel.</li> <li>■ In that mode the RxEnDlyTg1 should be configured with the minimum across all lanes of the dbyte of the RxDigStrbDlyTg1.</li> </ul> <p><b>Note:</b> These Registers are replicated per pstate.  <b>Note:</b> Timing Group 1 = RANK1  <b>Value After Reset:</b> 0x100  <b>Exists:</b> Always</p>

### 13.1.23 TxDqsRightEyeOffsetTg0\_pX (for X = 0; X <= 3)

- **Description:** TxDqsRightEyeOffsetTg0\_pX: Write DQS Right Eye Offset (Timing Group 0).
- **Size:** 6 bits
- **Offset:** (0x10000+(j<<12))+0x22+(X\*0x100000)
- **Exists:** Always



**Table 13-27 Fields for Register: TxDqsRightEyeOffsetTg0\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
5:0	TxDqsRightEyeOffsetTg0_p0	R/W	<p>TxDqsRightEyeOffsetTg0_p0: Write DQS Offset (Timing Group 0). Applicable in ECC Mode for LPDDR5. Program with a difference of the Mid-Point value and the Right Edge.</p> <ul style="list-style-type: none"> <li>■ Tx DQs Offset ( 6b fine delays). (For Timing Group 0)</li> <li>■ TxDqsRightEyeOffsetTg0[5:0] is the fine (fractional UI) delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> </ul> <p><b>Note:</b> These Registers are replicated per pstate.  <b>Note:</b> Timing Group 0 = RANK0  <b>Value After Reset:</b> 0x1f  <b>Exists:</b> Always</p>

### 13.1.24 TxDqsRightEyeOffsetTg1\_pX (for X = 0; X <= 3)

- **Description:** TxDqsRightEyeOffsetTg1\_pX: Write DQS Right Eye Offset (Timing Group 1).
- **Size:** 6 bits
- **Offset:** (0x10000+(j<<12))+0x23+(X\*0x100000)
- **Exists:** Always

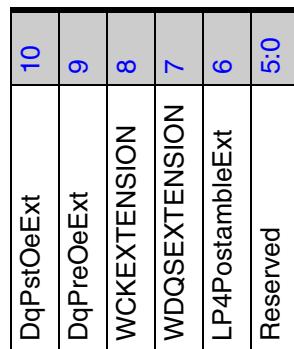


**Table 13-28 Fields for Register: TxDqsRightEyeOffsetTg1\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
5:0	TxDqsRightEyeOffsetTg1_p0	R/W	<p>TxDqsRightEyeOffsetTg1_p0: Write DQS Offset (Timing Group 1). Applicable in ECC Mode for LPDDR5. Program with a difference of the Mid-Point value and the Right Edge.</p> <ul style="list-style-type: none"> <li>■ Tx DQs Offset ( 6b fine delays). (For Timing Group 1)</li> <li>■ TxDqsRightEyeOffsetTg1[5:0] is the fine (fractional UI) delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> </ul> <p><b>Note:</b> These Registers are replicated per pstate.  <b>Note:</b> Timing Group 1 = RANK1  <b>Value After Reset:</b> 0x1f  <b>Exists:</b> Always</p>

### 13.1.25 DqsPreambleControl\_pX (for X = 0; X <= 3)

- **Name:** Control the PHY logic related to the read and write DQS preamble
- **Description:** DqsPreambleControl\_pX: For improved robustness in operating with DRAM timing drift, this register provides controls for the width of the write DQS preamble and postamble, and controls for the anticipated width of DQS preamble for reads.
- **Size:** 11 bits
- **Offset:** (0x10000+(j<<12))+0x24+(X\*0x100000)
- **Exists:** Always



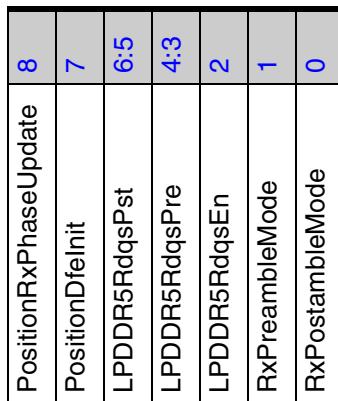
**Table 13-29 Fields for Register: DqsPreambleControl\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
10	DqPstOeExt	R/W	<p>DqPstOeExt: When set, OE is turned ON 1UI after the write burst for the DQ/DM pins of all the DBYTES. This Field should be set to 0 in LPDDR4X/5. Should be set to one for debug purposes only.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
9	DqPreOeExt	R/W	<p>DqPreOeExt: When set, DQ OE is turned ON 2UI early with a value of 0 for the DQ/DM pins for all the DBYTES. This Field should be set to 0 in LPDDR4X. Should be set to one for debug purposes only. This Field should be set to 1 in LPDDR5 Mode when DFE is supported by the DRAM which is based on MR24.OP[7]</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
8	WCkEXTENSION	R/W	<p>WCkEXTENSION: When set, WCK_T and WCK_C will be driven differentially to 0 and 1, respectively. This Field should be set to 0 in LPDDR4X. Should be set to one in LPDDR5 Mode for debug purposes only.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
7	WDQSEXTENSION	R/W	<p>WDQSEXTENSION: When set, DQS_T and DQS_C will be driven differentially to 0 and 1, respectively, before and after a read burst. See DesignWare Cores LPDDR4X MultiPHY: WDQS Extension Application Note. This Field should be set to 0 in LPDDR5 Mode.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
6	LP4PostambleExt	R/W	<p>LP4PostambleExt: In LPDDR4X mode must be set to extend the write postamble.</p> <ul style="list-style-type: none"> <li>■ 0: half-memclk write postamble</li> <li>■ 1: one-and-one-half-memclk write postamble; see LPDDR4X Spec MR3, OP[1] WR PST, vendor-specific function.</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
5:0	Reserved	R	<p>Reserved: Reserved</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.26 DbyteRxDqsModeCntrl\_pX (for X = 0; X <= 3)

- **Name:** Control for generating RxClk from read DQS edges.
- **Description:** DbyteRxDqsModeCntrl\_pX: Configures the read pipeline to use the DRAM generated read DQS.
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0x25+(X\*0x100000)
- **Exists:** Always



**Table 13-30 Fields for Register: DbyteRxDqsModeCntrl\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
8	PositionRxPhaseUpdate	R/W	PositionRxPhaseUpdate: Reserved. For firmware use only. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	PositionDfelnit	R/W	PositionDfelnit: Reserved. For firmware use only. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6:5	LPDDR5RdqsPst	R/W	LPDDR5RdqsPst: Configure with the same value as written to MR10 OP[7:6] RDQS PST <ul style="list-style-type: none"> <li>■ 00 is reserved</li> <li>■ 01 2.5*tWCK is supported,</li> <li>■ 10 4.5*tWCK is supported,</li> <li>■ 11 is reserved</li> </ul> <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

Bits	Name	Memory Access	Description
4:3	LPDDR5RdqsPre	R/W	<p>LPDDR5RdqsPre: Configure with the same value as written to MR10 OP[5:4] RDQS PRE</p> <ul style="list-style-type: none"> <li>■ 01 is the only supported value, Static 2*tWCK, Toggle 2*tWCK</li> <li>■ 00,10,11 are reserved</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
2	LPDDR5RdqsEn	R/W	<p>LPDDR5RdqsEn: Enables LPDDR5 RDQS mode. Must be 0 if EnStrbLssRdMode=1. Enables LPDDR5RdqsPre and LPDDR5RdqsPst. Disables RxPreambleMode and RxPostambleMode</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
1	RxPreambleMode	R/W	<p>RxPreambleMode: Reserved for LPDDR4X, unused in LPDDR5x_mphy.</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
0	RxPostambleMode	R/W	<p>RxPostambleMode: Reserved for LPDDR4X, unused in LPDDR5x_mphy.</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.1.27 RxClkCntl1\_pX (for X = 0; X <= 3)

- **Description:** RxClkCntl1\_pX: Controls for RxClk timing modes.
- **Size:** 1 bit
- **Offset:** (0x10000+(j<<12))+0x27+(X\*0x100000)
- **Exists:** Always



**Table 13-31 Fields for Register: RxClkCntl1\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
0	EnRxClkCor	R/W	<p>EnRxClkCor: Enables Correction term to be applied to RxClk based on RxReplica drift calculation</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.28 TxDqsDlyTg0\_pX (for X = 0; X <= 3)

- **Description:** TxDqsDlyTg0\_pX: Write DQS Delay (Timing Group DEST=0).
- **Size:** 10 bits
- **Offset:** (0x10000+(j<<12))+0x28+(X\*0x100000)
- **Exists:** Always

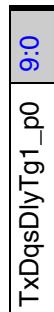


**Table 13-32 Fields for Register: TxDqsDlyTg0\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
9:0	TxDqsDlyTg0_p0	R/W	<p>TxDqsDlyTg0_p0: Write DQS Delay (Timing Group DEST=0). Trained to set the delay from the memory-write command to the signal driving the write DQS:</p> <ul style="list-style-type: none"> <li>■ Tx DQS Delay ( 10 bits, includes coarse and fine delays); (For Timing Group 0)</li> <li>■ TxDqsDlyTg0[9:6] is the coarse delay, i.e., one unit of delay is 1 UI.</li> <li>■ TxDqsDlyTg0[5:0] is the fine delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> </ul> <p>Register is per timing-group, per pstate.  <b>Note:</b> Timing Group 0 = RANK0  <b>Value After Reset:</b> 0x100  <b>Exists:</b> Always</p>

### 13.1.29 TxDqsDlyTg1\_pX (for X = 0; X <= 3)

- **Description:** TxDqsDlyTg1\_pX: Write DQS Delay (Timing Group DEST=1).
- **Size:** 10 bits
- **Offset:** (0x10000+(j<<12))+0x29+(X\*0x100000)
- **Exists:** Always

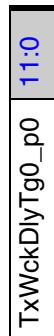


**Table 13-33 Fields for Register: TxDqsDlyTg1\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
9:0	TxDqsDlyTg1_p0	R/W	<p>TxDqsDlyTg1_p0: Write DQS Delay (Timing Group DEST=1). Trained to set the delay from the memory-write command to the signal driving the write DQS:</p> <ul style="list-style-type: none"> <li>■ Tx DQS Delay ( 10 bits, includes coarse and fine delays); (For Timing Group 1)</li> <li>■ TxDqsDlyTg1[9:6] is the coarse delay, i.e., one unit of delay is 1 UI.</li> <li>■ TxDqsDlyTg1[5:0] is the fine delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> </ul> <p>Register is per timing-group, per pstate.  <b>Note:</b> Timing Group 1 = RANK1  <b>Value After Reset:</b> 0x100  <b>Exists:</b> Always</p>

### 13.1.30 TxWckDlyTg0\_pX (for X = 0; X <= 3)

- **Description:** TxWckDlyTg0\_pX: Write WCK Delay (Timing Group DEST=0).
- **Size:** 12 bits
- **Offset:** (0x10000+(j<<12))+0x2a+(X\*0x100000)
- **Exists:** Always



**Table 13-34 Fields for Register: TxWckDlyTg0\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
11:0	TxWckDlyTg0_p0	R/W	<p>TxWckDlyTg0_p0: Write WCK Delay (Timing Group DEST=0). Trained to set the delay from the memory-write command to the signal driving the write WCK</p> <ul style="list-style-type: none"> <li>■ Tx WCK Delay ( 12 bits, includes coarse and fine delays); (For Timing Group 0)</li> <li>■ TxWckDlyTg0[11:6] is the coarse delay, i.e., one unit of delay is 1 UI.</li> </ul> <p>Max value supported for TxWckDlyTg0[11:6] is 26(decimal).</p> <ul style="list-style-type: none"> <li>■ TxWckDlyTg0[5:0] is the fine delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> </ul> <p>Register is per timing-group, per pstate.</p> <p><b>Note:</b> Timing Group 0 = RANK0 This CSR is applicable when design is compiled with DWC_LPDDR5XPHY_LPDDR5_ENABLED define.</p> <p><b>Value After Reset:</b> 0x200</p> <p><b>Exists:</b> Always</p>

### 13.1.31 TxWckDlyTg1\_pX (for X = 0; X <= 3)

- **Description:** TxWckDlyTg1\_pX: Write WCK Delay (Timing Group DEST=1).
- **Size:** 12 bits
- **Offset:** (0x10000+(j<<12))+0x2b+(X\*0x100000)
- **Exists:** Always



**Table 13-35 Fields for Register: TxWckDlyTg1\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
11:0	TxWckDlyTg1_p0	R/W	<p>TxWckDlyTg1_p0: Write WCK Delay (Timing Group DEST=1). Trained to set the delay from the memory-write command to the signal driving the write WCK</p> <ul style="list-style-type: none"> <li>■ Tx WCK Delay ( 12 bits, includes coarse and fine delays); (For Timing Group 1)</li> <li>■ TxWckDlyTg1[11:6] is the coarse delay, i.e., one unit of delay is 1 UI.</li> </ul> <p>Max value supported for TxWckDlyTg1[11:6] is 26(decimal).</p> <ul style="list-style-type: none"> <li>■ TxWckDlyTg1[5:0] is the fine delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> </ul> <p>Register is per timing-group, per pstate.</p> <p><b>Note:</b> Timing Group 1 = RANK1 This CSR is applicable when design is compiled with DWC_LPDDR5XPHY_LPDDR5_ENABLED define.</p> <p><b>Value After Reset:</b> 0x200</p> <p><b>Exists:</b> Always</p>

### 13.1.32 WrLevBits

- **Name:** Write level feedback DQ observability select.
- **Description:** WrLevBits: Write level feedback DQ observability selection
- **Size:** 8 bits
- **Offset:**  $(0x10000 + (j \ll 12)) + 0x2e$
- **Exists:** Always



**Table 13-36 Fields for Register: WrLevBits**

Bits	Name	Memory Access	Description
7:4	WrLevForDQSU	R/W	<p>WrLevForDQSU: Indicates which DQ bit is used for Write Levelization.</p> <ul style="list-style-type: none"> <li>■ 0000 - use DQ0</li> <li>■ ..</li> <li>■ 0111 - use DQ7</li> <li>■ 1000 - use IDQ[3:0]</li> <li>■ 1001 - use IDQ[7:4]</li> <li>■ 1010 - use IDQ[7:0]</li> </ul> <p><b>Value After Reset:</b> 0x9  <b>Exists:</b> Always</p>
3:0	WrLevForDQSL	R/W	<p>WrLevForDQSL: Indicates which DQ bit is used for Write Levelization.</p> <ul style="list-style-type: none"> <li>■ 0000 - use DQ0</li> <li>■ ..</li> <li>■ 0111 - use DQ7</li> <li>■ 1000 - use IDQ[3:0]</li> <li>■ 1001 - use IDQ[7:4]</li> <li>■ 1010 - use IDQ[7:0]</li> </ul> <p><b>Value After Reset:</b> 0x8  <b>Exists:</b> Always</p>

### 13.1.33 NeverGateDBDlyCalValClk

- **Description:** NeverGateDBDlyCalValClk: Reserved for PHY training firmware use.
- **Size:** 1 bit
- **Offset:** (0x10000+(j<<12))+0x2f
- **Exists:** Always

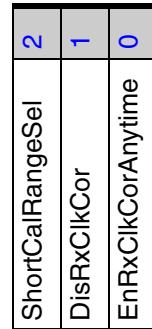


**Table 13-37 Fields for Register: NeverGateDBDlyCalValClk**

Bits	Name	Memory Access	Description
0	NeverGateDBDlyCalValClk	R/W	<p>NeverGateDBDlyCalValClk: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.34 RxClkCntl

- **Description:** RxClkCntl: Controls for RxClk timing modes.
- **Size:** 3 bits
- **Offset:** (0x10000+(j<<12))+0x31
- **Exists:** Always



**Table 13-38 Fields for Register: RxClkCntl**

Bits	Name	Memory Access	Description
2	ShortCalRangeSel	R/W	<p>ShortCalRangeSel: This field selects either of RxReplicaShortCalRangeA and RxReplicaShortCalRangeB value for the phase range we need to scan on both sides of the selected calibration phase</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
1	DisRxClkCor	R/W	<p>DisRxClkCor: Used to Dynamically Disable RxClk Timing updates.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
0	EnRxClkCorAnytime	R/W	<p>EnRxClkCorAnytime: This register is programmed as follows.</p> <ul style="list-style-type: none"> <li>■ 1: Enables the correction term to be applied to effective RxClkDly anytime and not just at time of update event.</li> <li>■ 0: Limits the update of effective RxClkDly to update events</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.35 RxModeCtlRxReplica\_pX (for X = 0; X <= 3)

- **Description:** RxModeCtlRxReplica\_pX: Deprecated
- **Size:** 4 bits
- **Offset:** (0x10000+(j<<12))+0x39+(X\*0x100000)
- **Exists:** Always



**Table 13-39 Fields for Register: RxModeCtlRxReplica\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
3:0	RxModeCtlRxReplica_p0	R/W	RxModeCtlRxReplica_p0: Deprecated <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.36 RxGainCurrAdjRxReplica\_pX (for X = 0; X <= 3)

- **Description:** RxGainCurrAdjRxReplica\_pX: Deprecated
- **Size:** 4 bits
- **Offset:** (0x10000+(j<<12))+0x3e+(X\*0x100000)
- **Exists:** Always



**Table 13-40 Fields for Register: RxGainCurrAdjRxReplica\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
3:0	RxGainCurrAdjRxReplica_p0	R/W	RxGainCurrAdjRxReplica_p0: Deprecated <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.37 TtcfControl

- **Name:** TTCF interface control
- **Description:** TtcfControl: Displays PHY Configuration
- **Size:** 2 bits
- **Offset:** (0x10000+(j<<12))+0x3f
- **Exists:** Always

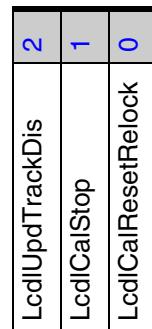


**Table 13-41 Fields for Register: TtcfControl**

Bits	Name	Memory Access	Description
1	TtcfForceHaltTransfers	R/W	<p>TtcfForceHaltTransfers: Halts all transfers across the TTCF interface while set to 1.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
0	TtcfForceSendAll	R/W	<p>TtcfForceSendAll: Setting this bit will cause all registers that are part of the TTCF interface to be marked as changed. This causes all values to be transferred over the TTCF interface.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.38 LcdlCalControl

- **Name:** DLL Lock State machine control register
- **Description:** LcdlCalControl:  
**Note:**This CSR is written by PHY Initialization Engine (PIE) and the data in here will be overwritten.
- **Size:** 3 bits
- **Offset:** (0x10000+(j<<12))+0x47
- **Exists:** Always



**Table 13-42 Fields for Register: LcdlCalControl**

Bits	Name	Memory Access	Description
2	LcdlUpdTrackDis	R/W	<p>LcdlUpdTrackDis: Disables LCDL calibration sequence on update events. Typically not used.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
1	LcdlCalStop	R/W	<p>LcdlCalStop: Pulsing high stops the locking sequence.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
0	LcdlCalResetRelock	R/W	<p>LcdlCalResetRelock: Pulsing high changes the LCDL state to unlocked. If the macro command FIFOs are not running, as when they are held in pointer init, then a seeded relock sequence begins immediately after pointer init when the command FIFOs are running. If the macro command FIFOs are running, then a seeded relock sequence begins immediately; this is not to be done usually.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.39 NeverGateTrainCntrClk

- **Description:** NeverGateTrainCntrClk: Reserved for PHY training firmware use.
- **Size:** 1 bit
- **Offset:** (0x10000+(j<<12))+0x49
- **Exists:** Always



**Table 13-43 Fields for Register: NeverGateTrainCntrClk**

Bits	Name	Memory Access	Description
0	NeverGateTrainCntrClk	R/W	<p>NeverGateTrainCntrClk: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.40 DBYTEParityInvert

- **Description:** DBYTEParityInvert: Invert APB Parity for register slave DBYTE
- **Size:** 2 bits
- **Offset:** (0x10000+(j<<12))+0x4d
- **Exists:** Always

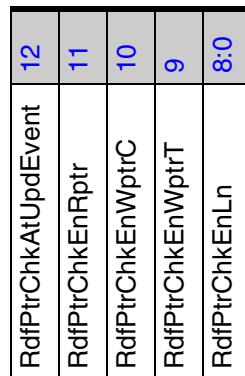


**Table 13-44 Fields for Register: DBYTEParityInvert**

Bits	Name	Memory Access	Description
1:0	DBYTEParityInvert	R/W	<p>DBYTEParityInvert: Invert APB Parity for register slave DBYTE. As required for Automotive. NOTE: This register should be used only for test. Set the bits for only one slave at a time. When bits are set for a particular slave. APB Reads of only that slave are valid. Bit 0 applies to [7:0] Bit 1 applies to [15:8] In HMZCAL, When ZcalStopClk=1, this csr can not be accessed.</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.1.41 RdfPtrChkControl

- **Name:** Controls for the checkers for the read-data-FIFOs read and write pointers.
- **Description:** RdfPtrChkControl: RDF pointer checker enables, per-lane and per type of pointer RDF pointer errors may cause an interrupt to the DFI controller; see CSRs PhyRdfPtrChkErrEn,Clr,Msk
- **Size:** 13 bits
- **Offset:** (0x10000+(j<<12))+0x5d
- **Exists:** Always



**Table 13-45 Fields for Register: RdfPtrChkControl**

Bits	Name	Memory Access	Description
12	RdfPtrChkAtUpdEvent	R/W	RdfPtrChkAtUpdEvent: Enables RDF pointer checks at phyupd and ctrlupd update events, aka LongBubble <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always
11	RdfPtrChkEnRptr	R/W	RdfPtrChkEnRptr: 1 enables RDF read pointer error checker 0 disables/clears Rptr errors <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always
10	RdfPtrChkEnWptrC	R/W	RdfPtrChkEnWptrC: 1 enables RDF write pointer C error checker 0 disables/clears WptrC errors <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always
9	RdfPtrChkEnWptrT	R/W	RdfPtrChkEnWptrT: 1 enables RDF write pointer T error checker 0 disables/clears WptrT errors <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always

Bits	Name	Memory Access	Description
8:0	RdfPtrChkEnLn	R/W	<p>RdfPtrChkEnLn: per-lane enable the checkers for the HM read-data FIFO pointers 1 enables the checkers for the HM read-data FIFO pointers for dq/se lane index 8 is for the dbi se lane in the upper nibble 0 disables/clears errors for lane of index.</p> <p><b>Value After Reset:</b> 0x1ff</p> <p><b>Exists:</b> Always</p>

### 13.1.42 DxRxStandbyEn\_pX (for X = 0; X <= 3)

- **Description:** DxRxStandbyEn\_pX: Per DBYTE RxStandby Control.
- **Size:** 1 bit
- **Offset:** (0x10000+(j<<12))+0x5f+(X\*0x100000)
- **Exists:** Always



**Table 13-46 Fields for Register: DxRxStandbyEn\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
0	DxRxStandbyEn_p0	R/W	<p>DxRxStandbyEn_p0: Per DBYTE RxStandby Control of the hard macro IO receiver circuits.</p> <ul style="list-style-type: none"> <li>■ 1: RxStandby is deasserted during read operations</li> <li>■ 1: otherwise asserted for power savings.</li> <li>■ 0: RxStandby is forced 0.</li> <li>■ using the csrs RxStandbyCntlByPState, RxPubLcdlSeed, RxPubCalModels1UI.</li> </ul> <p>Note: if the pstate0 value of csrRxPubCntlByPState = 0, then the pstate0 value of csrDxRxStandbyEn is used for all pstates.</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

### 13.1.43 TxDqLeftEyeOffsetTg0\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)

- **Description:** TxDqLeftEyeOffsetTg0\_rY\_pX: Write DQ Left Eye Offset (Timing Group 0).
- **Size:** 6 bits
- **Offset:** (0x10000+(j<<12))+0x60+(X\*0x100000)+(Y\*0x100)
- **Exists:** Always



**Table 13-47 Fields for Register: TxDqLeftEyeOffsetTg0\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)**

Bits	Name	Memory Access	Description
5:0	TxDqLeftEyeOffsetTg0_r0_p0	R/W	<p>TxDqLeftEyeOffsetTg0_r0_p0: Write DQ Offset (Timing Group 0). Program with a difference of the Mid-Point value and the Left Edge.</p> <ul style="list-style-type: none"> <li>■ Tx DQ Offset ( 6b fine delays). (For Timing Group 0)</li> <li>■ TxDqLeftEyeOffsetTg0[5:0] is the fine (fractional UI) delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> <li>■ Register Block Offset Address 0x0_1_0 contains the value for the lane0 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x1_1_0 contains the value for the lane1 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x2_1_0 contains the value for the lane2 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x3_1_0 contains the value for the lane3 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x4_1_0 contains the value for the lane4 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x5_1_0 contains the value for the lane5 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x6_1_0 contains the value for the lane6 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x7_1_0 contains the value for the lane7 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x8_1_0 contains the value for the lane[MDDBI] write DQ timing. (For Timing Group 0)</li> </ul> <p><b>Note:</b> These Registers are replicated per pstate.  <b>Note:</b> Timing Group 0 = RANK0  <b>Value After Reset:</b> 0x1f  <b>Exists:</b> Always</p>

### 13.1.44 TxDqLeftEyeOffsetTg1\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)

- **Description:** TxDqLeftEyeOffsetTg1\_rY\_pX: Write DQ Left Eye Offset (Timing Group 1).
- **Size:** 6 bits
- **Offset:**  $(0x10000 + (j \ll 12)) + 0x61 + (X * 0x100000) + (Y * 0x100)$
- **Exists:** Always



**Table 13-48 Fields for Register: TxDqLeftEyeOffsetTg1\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)**

Bits	Name	Memory Access	Description
5:0	TxDqLeftEyeOffsetTg1_r0_p0	R/W	<p>TxDqLeftEyeOffsetTg1_r0_p0: Write DQ Offset (Timing Group 1). Program with a difference of the Mid-Point value and the Left Edge.</p> <ul style="list-style-type: none"> <li>■ Tx DQ Offset ( 6b fine delays). (For Timing Group 1)</li> <li>■ TxDqLeftEyeOffsetTg1[5:0] is the fine (fractional UI) delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> <li>■ Register Block Offset Address 0x0_1_1 contains the value for the lane0 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x1_1_1 contains the value for the lane1 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x2_1_1 contains the value for the lane2 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x3_1_1 contains the value for the lane3 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x4_1_1 contains the value for the lane4 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x5_1_1 contains the value for the lane5 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x6_1_1 contains the value for the lane6 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x7_1_1 contains the value for the lane7 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x8_1_1 contains the value for the lane[MDDBI] write DQ timing. (For Timing Group 1)</li> </ul> <p><b>Note:</b> These Registers are replicated per pstate.  <b>Note:</b> Timing Group 1 = RANK1  <b>Value After Reset:</b> 0x1f  <b>Exists:</b> Always</p>

### 13.1.45 TxDqRightEyeOffsetTg0\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)

- **Description:** TxDqRightEyeOffsetTg0\_rY\_pX: Write DQ Right Eye Offset (Timing Group 0).
- **Size:** 6 bits
- **Offset:** (0x10000+(j<<12))+0x63+(X\*0x100000)+(Y\*0x100)
- **Exists:** Always



**Table 13-49 Fields for Register: TxDqRightEyeOffsetTg0\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)**

Bits	Name	Memory Access	Description
5:0	TxDqRightEyeOffsetTg0_r0_p0	R/W	<p>TxDqRightEyeOffsetTg0_r0_p0: Write DQ Offset (Timing Group 0). Program with a difference of the Mid-Point value and the Right Edge.</p> <ul style="list-style-type: none"> <li>■ Tx DQ Offset ( 6b fine delays). (For Timing Group 0)</li> <li>■ TxDqRightEyeOffsetTg0[5:0] is the fine (fractional UI) delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> <li>■ Register Block Offset Address 0x0_1_3 contains the value for the lane0 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x1_1_3 contains the value for the lane1 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x2_1_3 contains the value for the lane2 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x3_1_3 contains the value for the lane3 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x4_1_3 contains the value for the lane4 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x5_1_3 contains the value for the lane5 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x6_1_3 contains the value for the lane6 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x7_1_3 contains the value for the lane7 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x8_1_3 contains the value for the lane[DMDBI] write DQ timing. (For Timing Group 0)</li> </ul> <p><b>Note:</b>These Registers are replicated per pstate.  <b>Note:</b> Timing Group 0 = RANK0  <b>Value After Reset:</b> 0x1f  <b>Exists:</b> Always</p>

### 13.1.46 TxDqRightEyeOffsetTg1\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)

- **Description:** TxDqRightEyeOffsetTg1\_rY\_pX: Write DQ Right Eye Offset (Timing Group 1).
- **Size:** 6 bits
- **Offset:** (0x10000+(j<<12))+0x64+(X\*0x100000)+(Y\*0x100)
- **Exists:** Always



**Table 13-50 Fields for Register: TxDqRightEyeOffsetTg1\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)**

Bits	Name	Memory Access	Description
5:0	TxDqRightEyeOffsetTg1_r0_p0	R/W	<p>TxDqRightEyeOffsetTg1_r0_p0: Write DQ Offset (Timing Group 1). Program with a difference of the Mid-Point value and the Right Edge.</p> <ul style="list-style-type: none"> <li>■ Tx DQ Offset ( 6b fine delays). (For Timing Group 1)</li> <li>■ TxDqRightEyeOffsetTg1[5:0] is the fine (fractional UI) delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> <li>■ Register Block Offset Address 0x0_1_4 contains the value for the lane0 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x1_1_4 contains the value for the lane1 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x2_1_4 contains the value for the lane2 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x3_1_4 contains the value for the lane3 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x4_1_4 contains the value for the lane4 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x5_1_4 contains the value for the lane5 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x6_1_4 contains the value for the lane6 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x7_1_4 contains the value for the lane7 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x8_1_4 contains the value for the lane[DMDBI] write DQ timing. (For Timing Group 1)</li> </ul> <p><b>Note:</b>These Registers are replicated per pstate.  <b>Note:</b> Timing Group 1 = RANK1  <b>Value After Reset:</b> 0x1f  <b>Exists:</b> Always</p>

### 13.1.47 TrainingCntrSnap\_rX (for X = 0; X <= 8)

- **Description:** TrainingCntrSnap\_rX: Capture current contents of certain training registers.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0x65+(X\*0x100)
- **Exists:** Always



**Table 13-51 Fields for Register: TrainingCntrSnap\_rX (for X = 0; X <= 8)**

Bits	Name	Memory Access	Description
15:0	TrainingCntrSnap_r0	R	<p>TrainingCntrSnap_r0: The contents of the corresponding csrTrainingCntr is captured here on a 0 -&gt; 1 transition of csrTrainingResultsSnap.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.48 DtsmErrCountSnap\_iX (for X = 0; X <= 8)

- **Description:** DtsmErrCountSnap\_iX: Capture current contents of certain training registers.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0x66+(X\*0x100)
- **Exists:** Always



**Table 13-52 Fields for Register: DtsmErrCountSnap\_iX (for X = 0; X <= 8)**

Bits	Name	Memory Access	Description
15:0	DtsmErrCountSnap_i0	R	<p>DtsmErrCountSnap_i0: The contents of the corresponding csrDtsmErrCount is captured here on a 0 -&gt; 1 transition of csrTrainingResultsSnap.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.49 DtsmGoodCountSnap\_iX (for X = 0; X <= 8)

- **Description:** DtsmGoodCountSnap\_iX: Capture current contents of certain training registers.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0x67+(X\*0x100)
- **Exists:** Always



**Table 13-53 Fields for Register: DtsmGoodCountSnap\_iX (for X = 0; X <= 8)**

Bits	Name	Memory Access	Description
15:0	DtsmGoodCountSnap_i0	R	<p>DtsmGoodCountSnap_i0: The contents of the corresponding csrDtsmGoodCount is captured here on a 0 -&gt; 1 transition of csrTrainingResultsSnap.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.50 RxClkTLeftEyeOffsetTg0\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)

- **Description:** RxClkTLeftEyeOffsetTg0\_rY\_pX: RxClkT Left Eye Offset (Timing Group 0).
- **Size:** 6 bits
- **Offset:**  $(0x10000 + (j \ll 12)) + 0x68 + (X * 0x100000) + (Y * 0x100)$
- **Exists:** Always



**Table 13-54 Fields for Register: RxClkTLeftEyeOffsetTg0\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)**

Bits	Name	Memory Access	Description
5:0	RxClkTLeftEyeOffsetTg0_r0_p0	R/W	<p>RxClkTLeftEyeOffsetTg0_r0_p0: Read DQS Offset (Timing Group 0). Program with a difference of the Mid-Point value and the Left Edge.</p> <ul style="list-style-type: none"> <li>■ Rx Clk Offset ( 6b fine delays). (For Timing Group 0)</li> <li>■ RxClkLeftEyeOffsetTg0[5:0] is the fine (fractional UI) delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> <li>■ Register Block Offset Address 0x0_6_8 contains the value for the lane0 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x1_6_8 contains the value for the lane1 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x2_6_8 contains the value for the lane2 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x3_6_8 contains the value for the lane3 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x4_6_8 contains the value for the lane4 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x5_6_8 contains the value for the lane5 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x6_6_8 contains the value for the lane6 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x7_6_8 contains the value for the lane7 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x8_6_8 contains the value for the lane[DMDBI] RxClk timing. (For Timing Group 0)</li> </ul> <p><b>Note:</b>These Registers are replicated per pstate.  <b>Note:</b> Timing Group 0 = RANK0  <b>Value After Reset:</b> 0x1f  <b>Exists:</b> Always</p>

### 13.1.51 RxClkTLeftEyeOffsetTg1\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)

- **Description:** RxClkTLeftEyeOffsetTg1\_rY\_pX: RxClkT Left Eye Offset (Timing Group 1).
- **Size:** 6 bits
- **Offset:**  $(0x10000 + (j \ll 12)) + 0x69 + (X * 0x100000) + (Y * 0x100)$
- **Exists:** Always



**Table 13-55 Fields for Register: RxClkTLeftEyeOffsetTg1\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)**

Bits	Name	Memory Access	Description
5:0	RxClkTLeftEyeOffsetTg1_r0_p0	R/W	<p>RxClkTLeftEyeOffsetTg1_r0_p0: Read DQS Offset (Timing Group 1). Program with a difference of the Mid-Point value and the Left Edge.</p> <ul style="list-style-type: none"> <li>■ Rx Clk Offset ( 6b fine delays). (For Timing Group 1)</li> <li>■ RxClkLeftEyeOffsetTg1[5:0] is the fine (fractional UI) delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> <li>■ Register Block Offset Address 0x0_6_9 contains the value for the lane0 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x1_6_9 contains the value for the lane1 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x2_6_9 contains the value for the lane2 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x3_6_9 contains the value for the lane3 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x4_6_9 contains the value for the lane4 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x5_6_9 contains the value for the lane5 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x6_6_9 contains the value for the lane6 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x7_6_9 contains the value for the lane7 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x8_6_9 contains the value for the lane[DMDBI] RxClk timing. (For Timing Group 1)</li> </ul> <p><b>Note:</b>These Registers are replicated per pstate.  <b>Note:</b> Timing Group 1 = RANK1  <b>Value After Reset:</b> 0x1f  <b>Exists:</b> Always</p>

### 13.1.52 RxClkTRightEyeOffsetTg0\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)

- **Description:** RxClkTRightEyeOffsetTg0\_rY\_pX: RxClkT Right Eye Offset (Timing Group 0).
- **Size:** 6 bits
- **Offset:** (0x10000+(j<<12))+0x6a+(X\*0x100000)+(Y\*0x100)
- **Exists:** Always



**Table 13-56 Fields for Register: RxClkTRightEyeOffsetTg0\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)**

Bits	Name	Memory Access	Description
5:0	RxClkTRightEyeOffsetTg0_r0_p0	R/W	<p>RxClkTRightEyeOffsetTg0_r0_p0: Read DQS Offset (Timing Group 0). Program with a difference of the Mid-Point value and the Right Edge.</p> <ul style="list-style-type: none"> <li>■ Rx Clk Offset ( 6b fine delays). (For Timing Group 0)</li> <li>■ RxClkRightEyeOffsetTg0[5:0] is the fine (fractional UI) delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> <li>■ Register Block Offset Address 0x0_6_a contains the value for the lane0 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x1_6_a contains the value for the lane1 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x2_6_a contains the value for the lane2 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x3_6_a contains the value for the lane3 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x4_6_a contains the value for the lane4 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x5_6_a contains the value for the lane5 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x6_6_a contains the value for the lane6 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x7_6_a contains the value for the lane7 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x8_6_a contains the value for the lane[DMDBI] RxClk timing. (For Timing Group 0)</li> </ul> <p><b>Note:</b>These Registers are replicated per pstate.  <b>Note:</b> Timing Group 0 = RANK0  <b>Value After Reset:</b> 0x1f  <b>Exists:</b> Always</p>

### 13.1.53 RxClkTRightEyeOffsetTg1\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)

- **Description:** RxClkTRightEyeOffsetTg1\_rY\_pX: RxClkT Right Eye Offset (Timing Group 1).
- **Size:** 6 bits
- **Offset:**  $(0x10000 + (j \ll 12)) + 0x6b + (X * 0x100000) + (Y * 0x100)$
- **Exists:** Always



**Table 13-57 Fields for Register: RxClkTRightEyeOffsetTg1\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)**

Bits	Name	Memory Access	Description
5:0	RxClkTRightEyeOffsetTg1_r0_p0	R/W	<p>RxClkTRightEyeOffsetTg1_r0_p0: Read DQS Offset (Timing Group 1). Program with a difference of the Mid-Point value and the Right Edge.</p> <ul style="list-style-type: none"> <li>■ Rx Clk Offset ( 6b fine delays). (For Timing Group 1)</li> <li>■ RxClkRightEyeOffsetTg1[5:0] is the fine (fractional UI) delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> <li>■ Register Block Offset Address 0x0_6_b contains the value for the lane0 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x1_6_b contains the value for the lane1 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x2_6_b contains the value for the lane2 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x3_6_b contains the value for the lane3 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x4_6_b contains the value for the lane4 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x5_6_b contains the value for the lane5 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x6_6_b contains the value for the lane6 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x7_6_b contains the value for the lane7 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x8_6_b contains the value for the lane[DMDBI] RxClk timing. (For Timing Group 1)</li> </ul> <p><b>Note:</b>These Registers are replicated per pstate.  <b>Note:</b> Timing Group 1 = RANK1  <b>Value After Reset:</b> 0x1f  <b>Exists:</b> Always</p>

### 13.1.54 RxClkCLeftEyeOffsetTg0\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)

- **Description:** RxClkCLeftEyeOffsetTg0\_rY\_pX: RxClkC Left Eye Offset (Timing Group 0).
- **Size:** 6 bits
- **Offset:** (0x10000+(j<<12))+0x6c+(X\*0x100000)+(Y\*0x100)
- **Exists:** Always



**Table 13-58 Fields for Register: RxClkCLeftEyeOffsetTg0\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)**

Bits	Name	Memory Access	Description
5:0	RxClkCLeftEyeOffsetTg0_r0_p0	R/W	<p>RxClkCLeftEyeOffsetTg0_r0_p0: Read DQS Offset (Timing Group 0). Program with a difference of the Mid-Point value and the Left Edge.</p> <ul style="list-style-type: none"> <li>■ Rx Clk Offset ( 6b fine delays). (For Timing Group 0)</li> <li>■ RxClkLeftEyeOffsetTg0[5:0] is the fine (fractional UI) delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> <li>■ Register Block Offset Address 0x0_6_c contains the value for the lane0 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x1_6_c contains the value for the lane1 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x2_6_c contains the value for the lane2 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x3_6_c contains the value for the lane3 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x4_6_c contains the value for the lane4 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x5_6_c contains the value for the lane5 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x6_6_c contains the value for the lane6 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x7_6_c contains the value for the lane7 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x8_6_c contains the value for the lane[DMDBI] RxClk timing. (For Timing Group 0)</li> </ul> <p><b>Note:</b>These Registers are replicated per pstate.  <b>Note:</b> Timing Group 0 = RANK0  <b>Value After Reset:</b> 0x1f  <b>Exists:</b> Always</p>

### 13.1.55 RxClkCLeftEyeOffsetTg1\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)

- **Description:** RxClkCLeftEyeOffsetTg1\_rY\_pX: RxClkC Left Eye Offset (Timing Group 1).
- **Size:** 6 bits
- **Offset:** (0x10000+(j<<12))+0x6d+(X\*0x100000)+(Y\*0x100)
- **Exists:** Always



**Table 13-59 Fields for Register: RxClkCLeftEyeOffsetTg1\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)**

Bits	Name	Memory Access	Description
5:0	RxClkCLeftEyeOffsetTg1_r0_p0	R/W	<p>RxClkCLeftEyeOffsetTg1_r0_p0: Read DQS Offset (Timing Group 1). Program with a difference of the Mid-Point value and the Left Edge.</p> <ul style="list-style-type: none"> <li>■ Rx Clk Offset ( 6b fine delays). (For Timing Group 1)</li> <li>■ RxClkLeftEyeOffsetTg1[5:0] is the fine (fractional UI) delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> <li>■ Register Block Offset Address 0x0_6_d contains the value for the lane0 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x1_6_d contains the value for the lane1 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x2_6_d contains the value for the lane2 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x3_6_d contains the value for the lane3 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x4_6_d contains the value for the lane4 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x5_6_d contains the value for the lane5 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x6_6_d contains the value for the lane6 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x7_6_d contains the value for the lane7 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x8_6_d contains the value for the lane[DMDBI] RxClk timing. (For Timing Group 1)</li> </ul> <p><b>Note:</b>These Registers are replicated per pstate.  <b>Note:</b> Timing Group 1 = RANK1  <b>Value After Reset:</b> 0x1f  <b>Exists:</b> Always</p>

### 13.1.56 RxClkCRightEyeOffsetTg0\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)

- **Description:** RxClkCRightEyeOffsetTg0\_rY\_pX: RxClkC Right Eye Offset (Timing Group 0).
- **Size:** 6 bits
- **Offset:**  $(0x10000 + (j \ll 12)) + 0x6e + (X * 0x100000) + (Y * 0x100)$
- **Exists:** Always



**Table 13-60 Fields for Register: RxClkCRightEyeOffsetTg0\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)**

Bits	Name	Memory Access	Description
5:0	RxClkCRightEyeOffsetTg0_r0_p0	R/W	<p>RxClkCRightEyeOffsetTg0_r0_p0: Read DQS Offset (Timing Group 0). Program with a difference of the Mid-Point value and the Right Edge.</p> <ul style="list-style-type: none"> <li>■ Rx Clk Offset ( 6b fine delays). (For Timing Group 0)</li> <li>■ RxClkRightEyeOffsetTg0[5:0] is the fine (fractional UI) delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> <li>■ Register Block Offset Address 0x0_6_e contains the value for the lane0 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x1_6_e contains the value for the lane1 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x2_6_e contains the value for the lane2 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x3_6_e contains the value for the lane3 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x4_6_e contains the value for the lane4 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x5_6_e contains the value for the lane5 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x6_6_e contains the value for the lane6 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x7_6_e contains the value for the lane7 RxClk timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x8_6_e contains the value for the lane[DMDBI] RxClk timing. (For Timing Group 0)</li> </ul> <p><b>Note:</b>These Registers are replicated per pstate.  <b>Note:</b> Timing Group 0 = RANK0  <b>Value After Reset:</b> 0x1f  <b>Exists:</b> Always</p>

### 13.1.57 RxClkCRightEyeOffsetTg1\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)

- **Description:** RxClkCRightEyeOffsetTg1\_rY\_pX: RxClkC Right Eye Offset (Timing Group 1).
- **Size:** 6 bits
- **Offset:**  $(0x10000 + (j \ll 12)) + 0x6f + (X * 0x100000) + (Y * 0x100)$
- **Exists:** Always



**Table 13-61 Fields for Register: RxClkCRightEyeOffsetTg1\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)**

Bits	Name	Memory Access	Description
5:0	RxClkCRightEyeOffsetTg1_r0_p0	R/W	<p>RxClkCRightEyeOffsetTg1_r0_p0: Read DQS Offset (Timing Group 1). Program with a difference of the Mid-Point value and the Right Edge.</p> <ul style="list-style-type: none"> <li>■ Rx Clk Offset ( 6b fine delays). (For Timing Group 1)</li> <li>■ RxClkRightEyeOffsetTg1[5:0] is the fine (fractional UI) delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> <li>■ Register Block Offset Address 0x0_6_f contains the value for the lane0 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x1_6_f contains the value for the lane1 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x2_6_f contains the value for the lane2 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x3_6_f contains the value for the lane3 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x4_6_f contains the value for the lane4 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x5_6_f contains the value for the lane5 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x6_6_f contains the value for the lane6 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x7_6_f contains the value for the lane7 RxClk timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x8_6_f contains the value for the lane[DMDBI] RxClk timing. (For Timing Group 1)</li> </ul> <p><b>Note:</b>These Registers are replicated per pstate.  <b>Note:</b> Timing Group 1 = RANK1  <b>Value After Reset:</b> 0x1f  <b>Exists:</b> Always</p>

### 13.1.58 RxFifoVisibility

- **Name:** RX FIFO visibility
- **Description:** RxFifoVisibility: Allows the contents of the data rx FIFO to be read via CSRs RxDfe(0,1)FifoContentsLn(0..8). The FIFO must be quiescent, i.e., no actual operations in progress, while reading the FIFO. That is, the read from the RxFifo must be after the max-read-latency (DFIMRL) time. Procedure to read the contents of the entire RxFifo:
  - wait for any outstanding reads to complete (min 2\*DFIMRL MemClks)
  - read and store RxFifoWrLocEvnLn<1>[4:0] and RxFifoWrLocOddLn<1>[4:0] (they should be equal).
  - perform one BL16 memory reads. This will fill up 16 UI.
  - wait at least DFIMRL DfiClks
  - (loop 4 times to read 4 ui in each loop i.e., total 16 UI.)
  - **for (iNib=0, iNib<4, iNib++)**
  - - myRdPtr[5:0] = RxDfe0/1FifoRdLoc - 12 + iNib\*4 (WrEvn/OddPtr is 0 to 23 and RdPtr 0 to 47, so wrapping will have to be handled)
  - - write addr=RxFifoVisibility, data=[csrRxFifoVisRdEn=0 ,csrRxFifoVisRdPtr=myRdPtr]
  - - write addr=RxFifoVisibility, data=[csrRxFifoVisRdEn=1 ,csrRxFifoVisRdPtr=myRdPtr]
  - - write addr=RxFifoVisibility, data=[csrRxFifoVisRdEn=0 ,csrRxFifoVisRdPtr=myRdPtr]
  - - wait for 8 DfiClk.
  - - read addr=RxDfe0/1FifoContentsLn<8..0>
  - - unpack/align resulting data as described in descriptions of RxFifoContents registers, and
  - - map according to the configuration of the csr Dq<7..0>LnSel[2:0].
  - **end for**
  - evaluate/compare with data that travels through normal read path (dfi\_read\_data\_dbi,dfi\_read\_data).
  - write [csrRxFifoVisRdEn=0,csrRxFifoVisRdPtr = 0]
  - PHY is now available for normal mission-mode operation. end of Procedure
- **Size:** 8 bits
- **Offset:** (0x10000+(j<<12))+0x72
- **Exists:** Always



**Table 13-62 Fields for Register: RxFifoVisibility**

<b>Bits</b>	<b>Name</b>	<b>Memory Access</b>	<b>Description</b>
7	RxFifoVisRdEn	R/W	<p>RxFifoVisRdEn: Set 1 in this bit to read 4 or 8 UI of the DFE0 Fifo Contents in corresponding RxDfe0FifoContentsLn csr. (# of UI depends on DxCmdFifoWrMode)</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
6:0	RxFifoVisRdPtr	R/W	<p>RxFifoVisRdPtr: This 6b field addresses 8b units of the 72-entry FIFO. Valid values are 0..71.</p> <p>For example, Register RxFifoVisRdPtr[6:0]=7'd47 enables reading bit-entries 47 to 54.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.59 RxFifoContents\_rX (for X = 0; X <= 8)

- **Description:** RxFifoContents\_rX: Per Lane RX FIFO contents
- **Size:** 8 bits
- **Offset:** (0x10000+(j<<12))+0x73+(X\*0x100)
- **Exists:** Always



**Table 13-63 Fields for Register: RxFifoContents\_rX (for X = 0; X <= 8)**

Bits	Name	Memory Access	Description
7:0	RxFifoContents_r0	R	<p>RxFifoContents_r0: Contents of the RxFifo, as controlled by CSR RxFifoVisibility. This register reads max 8UI at a time from lane from the FIFO entries addressed by rdfifo_nibble_address[2:0]=RxFifoVisRdPtr[2:0]. Register[7:0] = [lane_ui7, lane_ui6... lane_ui0] where lane_ui0 is the first occurring bit</p> <p><b>Note:</b> The DBYTE lane of a given index is not the same as a memory DQ of the same index unless the csr Dq&lt;7..0&gt;LnSel[2:0] have their default/reset value.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.60 TrainingCntr\_rX (for X = 0; X <= 8)

- **Description:** TrainingCntr\_rX: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0x74+(X\*0x100)
- **Exists:** Always

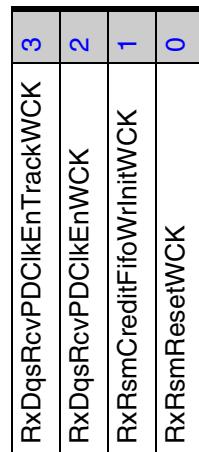


**Table 13-64 Fields for Register: TrainingCntr\_rX (for X = 0; X <= 8)**

Bits	Name	Memory Access	Description
15:0	TrainingCntr_r0	R/W	TrainingCntr_r0: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.61 WckDiffCtl

- **Name:** CSR control for WCK DIFF slices
- **Description:** WckDiffCtl: CSR Control for WCK DIFF Slices
- **Size:** 4 bits
- **Offset:** (0x10000+(j<<12))+0x75
- **Exists:** Always



**Table 13-65 Fields for Register: WckDiffCtl**

Bits	Name	Memory Access	Description
3	RxDqsRcvPDClkEnTrackWCK	R/W	RxDqsRcvPDClkEnTrackWCK: Control RxDqsRcvPDClkEn_train port of the WCK Diff Slice. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
2	RxDqsRcvPDClkEnWCK	R/W	RxDqsRcvPDClkEnWCK: Control RxDqsRcvPDClkEn_train port of the WCK Diff Slice. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
1	RxRsmCreditFifoWrInitWCK	R/W	RxRsmCreditFifoWrInitWCK: Control RxRsmCreditFifoWrInit port of the WCK Diff Slice. <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always
0	RxRsmResetWCK	R/W	RxRsmResetWCK: Control RxRsmReset port of the WCK Diff Slice. <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always

### 13.1.62 PptRxClkInfo\_rX (for X = 0; X <= 8)

- **Description:** PptRxClkInfo\_rX: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0x77+(X\*0x100)
- **Exists:** Always



**Table 13-66 Fields for Register: PptRxClkInfo\_rX (for X = 0; X <= 8)**

Bits	Name	Memory Access	Description
15:0	PptRxClkInfo_r0	R	PptRxClkInfo_r0: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always <b>Reset Mask:</b> 0x0 <b>Volatile:</b> true

### 13.1.63 RxDigStrbDlyTg0\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)

- **Description:** RxDigStrbDlyTg0\_rY\_pX: Rx Digital Strobe Coarse Delay (Timing Group 0).
- **Size:** 12 bits
- **Offset:** (0x10000+(j<<12))+0x78+(X\*0x100000)+(Y\*0x100)
- **Exists:** Always



**Table 13-67 Fields for Register: RxDigStrbDlyTg0\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)**

Bits	Name	Memory Access	Description
11:0	RxDigStrbDlyTg0_r0_p0	R/W	<p>RxDigStrbDlyTg0_r0_p0: Used when EnStrblssRdMode=1, otherwise unused. RxDigStrobe Delay (Timing Group 0) in units of 1/64 UI Maximum value is 12'h9ff = 39 + 63/64 UI Trained to center the nth RxDigStrobe in the first DQ eye.</p> <ul style="list-style-type: none"> <li>■ RxDigStrbDlyTg0[11:7] is the delay provided by the LP5 Rx pipe, in units of 2 UI.</li> <li>■ RxDigStrbDlyTg0[6:0] is the delay by the hardmacro LCDL circuit, in units of 1/64 UI.</li> </ul> <p>See FieldDescription of csrRxEnDlyTg for required cooperating configuration; that is, it must be configured with the minimum across all lanes of the values of the RxDigStrbDlyTg in this dbyte,. CsrRxPubCalModels1UI must be configured with the same value as register HMRxLcdlSeed csr RxCalModels1UI.</p> <ul style="list-style-type: none"> <li>■ Register Block Offset Address 0x0_7_8 controls the lane0 RxDigStrobe timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x1_7_8 controls the lane1 RxDigStrobe timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x2_7_8 controls the lane2 RxDigStrobe timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x3_7_8 controls the lane3 RxDigStrobe timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x4_7_8 controls the lane4 RxDigStrobe timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x5_7_8 controls the lane5 RxDigStrobe timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x6_7_8 controls the lane6 RxDigStrobe timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x7_7_8 controls the lane7 RxDigStrobe timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x8_7_8 controls the lane[DMDBI] RxDigStrobe timing. (For Timing Group 0)</li> </ul> <p><b>Note:</b> These Registers are replicated per pstate.  <b>Note:</b> Timing Group 0 = RANK0  <b>Value After Reset:</b> 0x100  <b>Exists:</b> Always</p>

### 13.1.64 RxDigStrbDlyTg1\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)

- **Description:** RxDigStrbDlyTg1\_rY\_pX: Rx Digital Strobe Coarse Delay (Timing Group 1).
- **Size:** 12 bits
- **Offset:** (0x10000+(j<<12))+0x79+(X\*0x100000)+(Y\*0x100)
- **Exists:** Always



**Table 13-68 Fields for Register: RxDigStrbDlyTg1\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)**

Bits	Name	Memory Access	Description
11:0	RxDigStrbDlyTg1_r0_p0	R/W	<p>RxDigStrbDlyTg1_r0_p0: Used when EnStrblssRdMode=1, otherwise unused. RxDigStrobe Delay (Timing Group 1) in units of 1/64 UI Maximum value is 12'h9ff = 39 + 63/64 UI Trained to center the nth RxDigStrobe in the first DQ eye.</p> <ul style="list-style-type: none"> <li>■ RxDigStrbDlyTg1[11:7] is the delay provided by the LP5 Rx pipe, in units of 2 UI.</li> <li>■ RxDigStrbDlyTg1[6:0] is the delay by the hardmacro LCDL circuit, in units of 1/64 UI.</li> </ul> <p>See FieldDescription of csrRxEnDlyTg for required cooperating configuration; that is, it must be configured with the minimum across all lanes of the values of the RxDigStrbDlyTg in this dbyte,. CsrRxPubCalModels1UI must be configured with the same value as register HMRxLcdlSeed csr RxCalModels1UI.</p> <ul style="list-style-type: none"> <li>■ Register Block Offset Address 0x0_7_9 controls the lane0 RxDigStrobe timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x1_7_9 controls the lane1 RxDigStrobe timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x2_7_9 controls the lane2 RxDigStrobe timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x3_7_9 controls the lane3 RxDigStrobe timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x4_7_9 controls the lane4 RxDigStrobe timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x5_7_9 controls the lane5 RxDigStrobe timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x6_7_9 controls the lane6 RxDigStrobe timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x7_7_9 controls the lane7 RxDigStrobe timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x8_7_9 controls the lane[DMDBI] RxDigStrobe timing. (For Timing Group 1)</li> </ul> <p><b>Note:</b> These Registers are replicated per pstate.  <b>Note:</b> Timing Group 1 = RANK1  <b>Value After Reset:</b> 0x100  <b>Exists:</b> Always</p>

### 13.1.65 TxDqDlyTg0\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)

- **Description:** TxDqDlyTg0\_rY\_pX: Write DQ Delay (Timing Group 0).
- **Size:** 10 bits
- **Offset:** (0x10000+(j<<12))+0x7a+(X\*0x100000)+(Y\*0x100)
- **Exists:** Always



**Table 13-69 Fields for Register: TxDqDlyTg0\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)**

Bits	Name	Memory Access	Description
9:0	TxDqDlyTg0_r0_p0	R/W	<p>TxDqDlyTg0_r0_p0: Write DQ Delay (Timing Group 0). Trained to center the delay between the write DQ to the write DQS.</p> <ul style="list-style-type: none"> <li>■ Tx DQ Delay ( 10 bits, includes 4b coarse and 6b fine delays). (For Timing Group 0)</li> <li>■ TxDqDlyTg0[9:6] is the coarse delay, i.e., one unit of delay is 1 UI. Max coarse delay should not exceed 8 UI.</li> <li>■ TxDqDlyTg0[5:0] is the fine (fractional UI) delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> <li>■ Register Block Offset Address 0x0_7_a controls the lane0 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x1_7_a controls the lane1 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x2_7_a controls the lane2 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x3_7_a controls the lane3 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x4_7_a controls the lane4 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x5_7_a controls the lane5 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x6_7_a controls the lane6 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x7_7_a controls the lane7 write DQ timing. (For Timing Group 0)</li> <li>■ Register Block Offset Address 0x8_7_a controls the lane[DMDBI] write DQ timing. (For Timing Group 0)</li> </ul> <p><b>Note:</b> These Registers are replicated per pstate.  <b>Note:</b> Timing Group 0 = RANK0  <b>Value After Reset:</b> 0x20  <b>Exists:</b> Always</p>

### 13.1.66 TxDqDlyTg1\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)

- **Description:** TxDqDlyTg1\_rY\_pX: Write DQ Delay (Timing Group 1).
- **Size:** 10 bits
- **Offset:** (0x10000+(j<<12))+0x7b+(X\*0x100000)+(Y\*0x100)
- **Exists:** Always

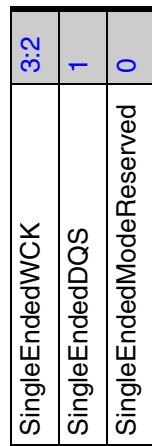


**Table 13-70 Fields for Register: TxDqDlyTg1\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)**

Bits	Name	Memory Access	Description
9:0	TxDqDlyTg1_r0_p0	R/W	<p>TxDqDlyTg1_r0_p0: Write DQ Delay (Timing Group 1). Trained to center the delay between the write DQ to the write DQS.</p> <ul style="list-style-type: none"> <li>■ Tx DQ Delay ( 10 bits, includes 4b coarse and 6b fine delays). (For Timing Group 1)</li> <li>■ TxDqDlyTg1[9:6] is the coarse delay, i.e., one unit of delay is 1 UI. Max coarse delay should not exceed 8 UI.</li> <li>■ TxDqDlyTg1[5:0] is the fine (fractional UI) delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> <li>■ Register Block Offset Address 0x0_7_b controls the lane0 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x1_7_b controls the lane1 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x2_7_b controls the lane2 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x3_7_b controls the lane3 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x4_7_b controls the lane4 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x5_7_b controls the lane5 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x6_7_b controls the lane6 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x7_7_b controls the lane7 write DQ timing. (For Timing Group 1)</li> <li>■ Register Block Offset Address 0x8_7_b controls the lane[DMDBI] write DQ timing. (For Timing Group 1)</li> </ul> <p><b>Note:</b>These Registers are replicated per pstate.  <b>Note:</b> Timing Group 1 = RANK1  <b>Value After Reset:</b> 0x20  <b>Exists:</b> Always</p>

### 13.1.67 SingleEndedMode\_pX (DBYTE) (for X = 0; X <= 3)

- **Name:** Control for Single Ended LP4X/5 Mode
- **Description:** SingleEndedMode\_pX: Configures the DIFF IO to operate in Single Ended Mode.
- **Size:** 4 bits
- **Offset:**  $(0x10000 + (j \ll 12)) + 0x7c + (X * 0x100000)$
- **Exists:** Always



**Table 13-71 Fields for Register: SingleEndedMode\_pX (DBYTE) (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
3:2	SingleEndedWCK	R/W	<p>SingleEndedWCK: This field is programmed as follows:</p> <ul style="list-style-type: none"> <li>■ Value of 2'b00 - Drives Differential value on TxDataT and TxDataC of WCK DIFF Slice. (Default)</li> <li>■ Value of 2'b01 - Drives TxDataC of WCK DIFF Slice to 0.</li> <li>■ Value of 2'b10 - Drives TxDataT of WCK DIFF Slice to 0.</li> <li>■ Value of 2'b11 - Reserved.</li> </ul> <p>This is applicable in LPDDR5 Mode.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
1	SingleEndedDQS	R/W	<p>SingleEndedDQS: Drives TxDataC of DQS DIFF Slice to 0.</p> <p>This is applicable in LPDDR4X Mode.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
0	SingleEndedModeReserved	R/W	<p>SingleEndedModeReserved: Reserved</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.68 ScratchPadDBYTE

- **Description:** ScratchPadDBYTE: ScratchPad for DBYTE
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0x7d
- **Exists:** Always

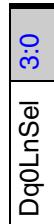


**Table 13-72 Fields for Register: ScratchPadDBYTE**

Bits	Name	Memory Access	Description
15:0	ScratchPadDBYTE	R/W	<p>ScratchPadDBYTE: ScratchPad for DBYTE. As required for Automotive. In HMZCAL, When ZcalStopClk=1, this csr can not be accessed.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.69 Dq0LnSel

- **Description:** Dq0LnSel: Maps Phy DQ lane to memory DQ0
- **Size:** 4 bits
- **Offset:**  $(0x10000 + (j << 12)) + 0x80$
- **Exists:** Always

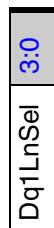


**Table 13-73 Fields for Register: Dq0LnSel**

Bits	Name	Memory Access	Description
3:0	Dq0LnSel	R/W	<p>Dq0LnSel: Supports mapping of PHY dq to dram dq within a byte (swizzle) for LPDDR5/LPDDR4X. Intended to undo the swizzle of board-level phy-to-dram connections. such that MRR operations of binary counters may be return correct values.</p> <p><b>Note:</b>Valid value = 0..8. DQ0..DQ7 = 0..7. DM/DBI = 8;</p> <p><b>Note:</b>This register is per-dbyte such that the swizzle may be different per dbyte. Each register in a byte's set of DqLnSel must have a unique value within the set.</p> <p>For example, if, on this dbyte, PHY lane 3 is connected to memory dq0 (on this dbyte), then Register Dq0LnSel for this dbyte should be 3.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.70 Dq1LnSel

- **Description:** Dq1LnSel: Maps Phy DQ lane to memory DQ1
- **Size:** 4 bits
- **Offset:**  $(0x10000 + (j << 12)) + 0x81$
- **Exists:** Always

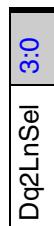


**Table 13-74 Fields for Register: Dq1LnSel**

Bits	Name	Memory Access	Description
3:0	Dq1LnSel	R/W	<p>Dq1LnSel: Supports mapping of PHY dq to dram dq within a byte (swizzle) for LPDDR5/LPDDR4X. Intended to undo the swizzle of board-level phy-to-dram connections. such that MRR operations of binary counters may be return correct values.</p> <p><b>Note:</b>Valid value = 0..8. DQ0..DQ7 = 0..7. DM/DBI = 8;</p> <p><b>Note:</b>This register is per-dbyte such that the swizzle may be different per dbyte. Each register in a byte's set of DqLnSel must have a unique value within the set.</p> <p>For example, if, on this dbyte, PHY lane 3 is connected to memory dq1 (on this dbyte), then Register Dq1LnSel for this dbyte should be 3.</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

### 13.1.71 Dq2LnSel

- **Description:** Dq2LnSel: Maps Phy DQ lane to memory DQ2
- **Size:** 4 bits
- **Offset:**  $(0x10000 + (j << 12)) + 0x82$
- **Exists:** Always

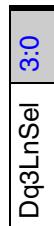


**Table 13-75 Fields for Register: Dq2LnSel**

Bits	Name	Memory Access	Description
3:0	Dq2LnSel	R/W	<p>Dq2LnSel: Supports mapping of PHY dq to dram dq within a byte (swizzle) for LPDDR5/LPDDR4X. Intended to undo the swizzle of board-level phy-to-dram connections. such that MRR operations of binary counters may be return correct values.</p> <p><b>Note:</b>Valid value = 0..8. DQ0..DQ7 = 0..7. DM/DBI = 8;</p> <p><b>Note:</b>This register is per-dbyte such that the swizzle may be different per dbyte. Each register in a byte's set of DqLnSel must have a unique value within the set.</p> <p>For example, if, on this dbyte, PHY lane 3 is connected to memory dq2 (on this dbyte), then Register Dq2LnSel for this dbyte should be 3.</p> <p><b>Value After Reset:</b> 0x2</p> <p><b>Exists:</b> Always</p>

### 13.1.72 Dq3LnSel

- **Description:** Dq3LnSel: Maps Phy DQ lane to memory DQ3
- **Size:** 4 bits
- **Offset:**  $(0x10000 + (j << 12)) + 0x83$
- **Exists:** Always

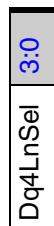


**Table 13-76 Fields for Register: Dq3LnSel**

Bits	Name	Memory Access	Description
3:0	Dq3LnSel	R/W	<p>Dq3LnSel: Supports mapping of PHY dq to dram dq within a byte (swizzle) for LPDDR5/LPDDR4X. Intended to undo the swizzle of board-level phy-to-dram connections. such that MRR operations of binary counters may be return correct values.</p> <p><b>Note:</b>Valid value = 0..8. DQ0..DQ7 = 0..7. DM/DBI = 8;</p> <p><b>Note:</b>This register is per-dbyte such that the swizzle may be different per dbyte. Each register in a byte's set of DqLnSel must have a unique value within the set.</p> <p>For example, if, on this dbyte, PHY lane 3 is connected to memory dq3 (on this dbyte), then Register Dq3LnSel for this dbyte should be 3.</p> <p><b>Value After Reset:</b> 0x3</p> <p><b>Exists:</b> Always</p>

### 13.1.73 Dq4LnSel

- **Description:** Dq4LnSel: Maps Phy DQ lane to memory DQ4
- **Size:** 4 bits
- **Offset:**  $(0x10000 + (j \ll 12)) + 0x84$
- **Exists:** Always

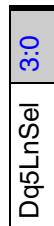


**Table 13-77 Fields for Register: Dq4LnSel**

Bits	Name	Memory Access	Description
3:0	Dq4LnSel	R/W	<p>Dq4LnSel: Supports mapping of PHY dq to dram dq within a byte (swizzle) for LPDDR5/LPDDR4X. Intended to undo the swizzle of board-level phy-to-dram connections. such that MRR operations of binary counters may be return correct values.</p> <p><b>Note:</b>Valid value = 0..8. DQ0..DQ7 = 0..7. DM/DBI = 8;</p> <p><b>Note:</b>This register is per-dbyte such that the swizzle may be different per dbyte. Each register in a byte's set of DqLnSel must have a unique value within the set.</p> <p>For example, if, on this dbyte, PHY lane 3 is connected to memory dq4 (on this dbyte), then Register Dq4LnSel for this dbyte should be 3.</p> <p><b>Value After Reset:</b> 0x4</p> <p><b>Exists:</b> Always</p>

### 13.1.74 Dq5LnSel

- **Description:** Dq5LnSel: Maps Phy DQ lane to memory DQ5
- **Size:** 4 bits
- **Offset:**  $(0x10000 + (j << 12)) + 0x85$
- **Exists:** Always

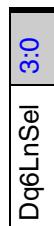


**Table 13-78 Fields for Register: Dq5LnSel**

Bits	Name	Memory Access	Description
3:0	Dq5LnSel	R/W	<p>Dq5LnSel: Supports mapping of PHY dq to dram dq within a byte (swizzle) for LPDDR5/LPDDR4X. Intended to undo the swizzle of board-level phy-to-dram connections. such that MRR operations of binary counters may be return correct values.</p> <p><b>Note:</b>Valid value = 0..8. DQ0..DQ7 = 0..7. DM/DBI = 8;</p> <p><b>Note:</b>This register is per-dbyte such that the swizzle may be different per dbyte. Each register in a byte's set of DqLnSel must have a unique value within the set.</p> <p>For example, if, on this dbyte, PHY lane 3 is connected to memory dq5 (on this dbyte), then Register Dq5LnSel for this dbyte should be 3.</p> <p><b>Value After Reset:</b> 0x5</p> <p><b>Exists:</b> Always</p>

### 13.1.75 Dq6LnSel

- **Description:** Dq6LnSel: Maps Phy DQ lane to memory DQ6
- **Size:** 4 bits
- **Offset:**  $(0x10000 + (j << 12)) + 0x86$
- **Exists:** Always

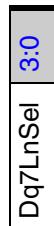


**Table 13-79 Fields for Register: Dq6LnSel**

Bits	Name	Memory Access	Description
3:0	Dq6LnSel	R/W	<p>Dq6LnSel: Supports mapping of PHY dq to dram dq within a byte (swizzle) for LPDDR5/LPDDR4X. Intended to undo the swizzle of board-level phy-to-dram connections. such that MRR operations of binary counters may be return correct values.</p> <p><b>Note:</b>Valid value = 0..8. DQ0..DQ7 = 0..7. DM/DBI = 8;</p> <p><b>Note:</b>This register is per-dbyte such that the swizzle may be different per dbyte. Each register in a byte's set of DqLnSel must have a unique value within the set.</p> <p>For example, if, on this dbyte, PHY lane 3 is connected to memory dq6 (on this dbyte), then Register Dq6LnSel for this dbyte should be 3.</p> <p><b>Value After Reset:</b> 0x6</p> <p><b>Exists:</b> Always</p>

### 13.1.76 Dq7LnSel

- **Description:** Dq7LnSel: Maps Phy DQ lane to memory DQ7
- **Size:** 4 bits
- **Offset:**  $(0x10000 + (j << 12)) + 0x87$
- **Exists:** Always

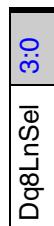


**Table 13-80 Fields for Register: Dq7LnSel**

Bits	Name	Memory Access	Description
3:0	Dq7LnSel	R/W	<p>Dq7LnSel: Supports mapping of PHY dq to dram dq within a byte (swizzle) for LPDDR5/LPDDR4X. Intended to undo the swizzle of board-level phy-to-dram connections. such that MRR operations of binary counters may be return correct values.</p> <p><b>Note:</b>Valid value = 0..8. DQ0..DQ7 = 0..7. DM/DBI = 8;</p> <p><b>Note:</b>This register is per-dbyte such that the swizzle may be different per dbyte. Each register in a byte's set of DqLnSel must have a unique value within the set.</p> <p>For example, if, on this dbyte, PHY lane 3 is connected to memory dq7 (on this dbyte), then Register Dq7LnSel for this dbyte should be 3.</p> <p><b>Value After Reset:</b> 0x7</p> <p><b>Exists:</b> Always</p>

### 13.1.77 Dq8LnSel

- **Description:** Dq8LnSel: Maps Phy DQ lane to memory DQ8
- **Size:** 4 bits
- **Offset:**  $(0x10000 + (j << 12)) + 0x88$
- **Exists:** Always



**Table 13-81 Fields for Register: Dq8LnSel**

Bits	Name	Memory Access	Description
3:0	Dq8LnSel	R/W	<p>Dq8LnSel: Supports mapping of PHY dq to dram dq within a byte (swizzle) for LPDDR5/LPDDR4X. Intended to undo the swizzle of board-level phy-to-dram connections. such that MRR operations of binary counters may be return correct values.</p> <p><b>Note:</b>Valid value = 0..8. DQ0..DQ7 = 0..7. DM/DBI = 8;</p> <p><b>Note:</b>This register is per-dbyte such that the swizzle may be different per dbyte. Each register in a byte's set of DqLnSel must have a unique value within the set.</p> <p>For example, if, on this dbyte, PHY lane 3 is connected to memory dq8 (on this dbyte), then Register Dq8LnSel for this dbyte should be 3.</p> <p><b>Value After Reset:</b> 0x8</p> <p><b>Exists:</b> Always</p>

### 13.1.78 AsyncDbyteTxMode

- **Description:** AsyncDbyteTxMode: Reserved for PHY training firmware use.
- **Size:** 13 bits
- **Offset:** (0x10000+(j<<12))+0x89
- **Exists:** Always



**Table 13-82 Fields for Register: AsyncDbyteTxMode**

Bits	Name	Memory Access	Description
12:0	AsyncDbyteTxMode	R/W	AsyncDbyteTxMode: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.79 AsyncDbyteRxMode

- **Description:** AsyncDbyteRxMode: Reserved for PHY training firmware use.
- **Size:** 11 bits
- **Offset:**  $(0x10000 + (j \ll 12)) + 0x8a$
- **Exists:** Always



**Table 13-83 Fields for Register: AsyncDbyteRxMode**

Bits	Name	Memory Access	Description
10:0	AsyncDbyteRxMode	R/W	AsyncDbyteRxMode: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.80 AsyncDbyteTxEn

- **Description:** AsyncDbyteTxEn: Reserved for PHY training firmware use.
- **Size:** 13 bits
- **Offset:** (0x10000+(j<<12))+0x8b
- **Exists:** Always



**Table 13-84 Fields for Register: AsyncDbyteTxEn**

Bits	Name	Memory Access	Description
12:0	AsyncDbyteTxEn	R/W	<p>AsyncDbyteTxEn: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.81 AsyncDbyteTxData

- **Description:** AsyncDbyteTxData: Reserved for PHY training firmware use.
- **Size:** 13 bits
- **Offset:** (0x10000+(j<<12))+0x8c
- **Exists:** Always



**Table 13-85 Fields for Register: AsyncDbyteTxData**

Bits	Name	Memory Access	Description
12:0	AsyncDbyteTxData	R/W	<p>AsyncDbyteTxData: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.82 AsyncDbyteRxData

- **Description:** AsyncDbyteRxData: Reserved for PHY training firmware use.
- **Size:** 13 bits
- **Offset:** (0x10000+(j<<12))+0x8d
- **Exists:** Always



**Table 13-86 Fields for Register: AsyncDbyteRxData**

Bits	Name	Memory Access	Description
12:0	AsyncDbyteRxData	R	<p>AsyncDbyteRxData: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.83 SelDbCurDlyTmngInfo

- **Description:** SelDbCurDlyTmngInfo: Reserved for PHY training firmware use and for debugging
- **Size:** 8 bits
- **Offset:** (0x10000+(j<<12))+0x8e
- **Exists:** Always



**Table 13-87 Fields for Register: SelDbCurDlyTmngInfo**

Bits	Name	Memory Access	Description
7:0	SelDbCurDlyTmngInfo	R/W	<p>SelDbCurDlyTmngInfo: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.84 DxLoopBackEn

- **Description:** DxLoopBackEn: Per DBYTE Loopback Enable Configuration
- **Size:** 1 bit
- **Offset:** (0x10000+(j<<12))+0x92
- **Exists:** Always



**Table 13-88 Fields for Register: DxLoopBackEn**

Bits	Name	Memory Access	Description
0	DxLoopBackEn	R/W	<p>DxLoopBackEn: This field is programmed as follows:</p> <ul style="list-style-type: none"> <li>■ Set this to 1 for PadSide Loopback in addition to DxCore-LoopBackMode.</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.1.85 DxDigStrobeGenSel

- **Description:** DxDigStrobeGenSel: Selects the source for the generation of the RxDigStrobe
- **Size:** 1 bit
- **Offset:** (0x10000+(j<<12))+0x94
- **Exists:** Always



**Table 13-89 Fields for Register: DxDigStrobeGenSel**

Bits	Name	Memory Access	Description
0	DxDigStrobeGenSel	R/W	<p>DxDigStrobeGenSel: This field is programmed as follows:</p> <ul style="list-style-type: none"> <li>■ 1'b1 - RxDigStrobe is Generated from CSR.</li> <li>■ 1'b0 - RxDigStrobe is Generated from Read Transaction.</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.86 DxDigStrobePat

- **Description:** DxDigStrobePat: Selects the pattern for RxDigStrobe
- **Size:** 8 bits
- **Offset:** (0x10000+(j<<12))+0x95
- **Exists:** Always



**Table 13-90 Fields for Register: DxDigStrobePat**

Bits	Name	Memory Access	Description
7:0	DxDigStrobePat	R/W	<p>DxDigStrobePat: 8 UI Pattern for the generation RxDigStrobe. This pattern will be set on the RxDigStrobe if DxDigStrobeGenSel = 1'b1</p> <p><b>Value After Reset:</b> Oxaa</p> <p><b>Exists:</b> Always</p>

### 13.1.87 DxRxStrobeEnPatWck

- **Description:** DxRxStrobeEnPatWck: Selects the pattern for RxStrobeEn for WCK DIFF Slice.
- **Size:** 8 bits
- **Offset:** (0x10000+(j<<12))+0x96
- **Exists:** Always

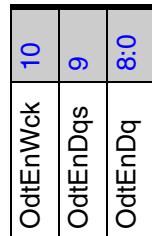


**Table 13-91 Fields for Register: DxRxStrobeEnPatWck**

Bits	Name	Memory Access	Description
7:0	DxRxStrobeEnPatWck	R/W	<p>DxRxStrobeEnPatWck: 8 UI Pattern for the generation RxStrobeEn</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.88 DxOdtEn

- **Name:** DBYTE OdtEn control
- **Description:** DxOdtEn: DQS DIFF Slice DBYTE Slice IOOdtEn Control OdtEn should be asserted for mission mode
- **Size:** 11 bits
- **Offset:** (0x10000+(j<<12))+0x97
- **Exists:** Always

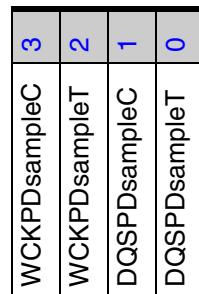


**Table 13-92 Fields for Register: DxOdtEn**

Bits	Name	Memory Access	Description
10	OdtEnWck	R/W	<p>OdtEnWck: Signal which controls WCK DIFF IO OdtEn</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
9	OdtEnDqs	R/W	<p>OdtEnDqs: Signal which controls DQS DIFF IO OdtEn. The per-pstate csrOdtDisDqs overrides this csr.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
8:0	OdtEnDq	R/W	<p>OdtEnDq: Signal which controls SE IO OdtEn.</p> <ul style="list-style-type: none"> <li>■ 7:0 -&gt; dq[7:0]</li> <li>■ 8 -&gt; dbi</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.89 DxPDsampleDIFF

- **Name:** Address Loopback sample values for Diff slice
- **Description:** DxPDsampleDIFF: This returns the diff slice's loopback value sampled by PDClk.
- **Size:** 4 bits
- **Offset:** (0x10000+(j<<12))+0x98
- **Exists:** Always



**Table 13-93 Fields for Register: DxPDsampleDIFF**

Bits	Name	Memory Access	Description
3	WCKPDsampleC	R	<p>WCKPDsampleC: PD sample value returned by Complementary bit of WCK Diff slice. This field is applicable when design is compiled with DWC_LPDDR5XPHY_LPDDR5_ENABLED define.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
2	WCKPDsampleT	R	<p>WCKPDsampleT: PD sample value returned by True bit of WCK Diff slice. This field is applicable when design is compiled with DWC_LPDDR5XPHY_LPDDR5_ENABLED define.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
1	DQSPDsampleC	R	<p>DQSPDsampleC: PD sample value returned by Complementary bit of DQS Diff slice</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

Bits	Name	Memory Access	Description
0	DQSPDsampleT	R	<p>DQSPDsampleT: PD sample value returned by True bit of DQS Diff slice</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.90 RxClkTrkErr00

- **Name:** For RxClkDly tracking of tPHY\_tDQS2DQ; overflow, underflow for Tg0,Tg1
- **Description:** RxClkTrkErr00: Arithmetic overflow, underflow status when correcting effective RxClk[T,C]Dly. Design assumptions about the range of VT variation violated. These status bits are the OR of the status of the DQ/DBI lanes. The Snap values are latched by the assertion of RxReplicaStatusSnapNow
- The Update values are latched by the update events. All values are reset to 0 by PUB Reset, by PtrInit or by DByteDisable.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0x99
- **Exists:** Always

15	RxClikTTTrkOverflowSnapTg1
14	RxClikTTTrkUnderflowSnapTg1
13	RxClikCTrkOverflowSnapTg1
12	RxClikCTrkUnderflowSnapTg1
11	RxClikTTTrkOverflowSnapTg0
10	RxClikTTTrkUnderflowSnapTg0
9	RxClikCTrkOverflowSnapTg0
8	RxClikCTrkUnderflowSnapTg0
7	RxClikTTTrkOverflowUpdateTg1
6	RxClikTTTrkUnderflowUpdateTg1
5	RxClikCTrkOverflowUpdateTg1
4	RxClikCTrkUnderflowUpdateTg1
3	RxClikTTTrkOverflowUpdateTg0
2	RxClikTrkUnderflowUpdateTg0
1	RxClikCTrkOverflowUpdateTg0
0	RxClikCTrkUnderflowUpdateTg0

Table 13-94 Fields for Register: RxClkTrkErr00

Bits	Name	Memory Access	Description
15	RxClikTTTrkOverflowSnapTg1	R	<p>RxClikTTTrkOverflowSnapTg1: For RxClkDly tracking of tPHY_tDQS2DQ; overflow, underflow for Tg0,Tg1</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
14	RxClikTTTrkUnderflowSnapTg1	R	<p>RxClikTTTrkUnderflowSnapTg1: For RxClkDly tracking of tPHY_tDQS2DQ; overflow, underflow for Tg0,Tg1</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

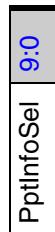
Bits	Name	Memory Access	Description
13	RxClkCTrkOverflowSnapTg1	R	<p>RxClkCTrkOverflowSnapTg1: For RxClkDly tracking of tPHY_tDQS2DQ; overflow, underflow for Tg0,Tg1</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
12	RxClkCTrkUnderflowSnapTg1	R	<p>RxClkCTrkUnderflowSnapTg1: For RxClkDly tracking of tPHY_tDQS2DQ; overflow, underflow for Tg0,Tg1</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
11	RxClkTTrkOverflowSnapTg0	R	<p>RxClkTTrkOverflowSnapTg0: For RxClkDly tracking of tPHY_tDQS2DQ; overflow, underflow for Tg0,Tg1</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
10	RxClktTrkUnderflowSnapTg0	R	<p>RxClktTrkUnderflowSnapTg0: For RxClkDly tracking of tPHY_tDQS2DQ; overflow, underflow for Tg0,Tg1</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
9	RxClkCTrkOverflowSnapTg0	R	<p>RxClkCTrkOverflowSnapTg0: For RxClkDly tracking of tPHY_tDQS2DQ; overflow, underflow for Tg0,Tg1</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
8	RxClkCTrkUnderflowSnapTg0	R	<p>RxClkCTrkUnderflowSnapTg0: For RxClkDly tracking of tPHY_tDQS2DQ; overflow, underflow for Tg0,Tg1</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

Bits	Name	Memory Access	Description
7	RxClkTTrkOverflowUpdateTg1	R	<p>RxClkTTrkOverflowUpdateTg1: For RxClkDly tracking of tPHY_tDQS2DQ; overflow, underflow for Tg0,Tg1</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
6	RxClkTTrkUnderflowUpdateTg1	R	<p>RxClkTTrkUnderflowUpdateTg1: For RxClkDly tracking of tPHY_tDQS2DQ; overflow, underflow for Tg0,Tg1</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
5	RxClkCTrkOverflowUpdateTg1	R	<p>RxClkCTrkOverflowUpdateTg1: For RxClkDly tracking of tPHY_tDQS2DQ; overflow, underflow for Tg0,Tg1</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
4	RxClkCTrkUnderflowUpdateTg1	R	<p>RxClkCTrkUnderflowUpdateTg1: For RxClkDly tracking of tPHY_tDQS2DQ; overflow, underflow for Tg0,Tg1</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
3	RxClkTTrkOverflowUpdateTg0	R	<p>RxClkTTrkOverflowUpdateTg0: For RxClkDly tracking of tPHY_tDQS2DQ; overflow, underflow for Tg0,Tg1</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
2	RxClikTrkUnderflowUpdateTg0	R	<p>RxClikTrkUnderflowUpdateTg0: For RxClkDly tracking of tPHY_tDQS2DQ; overflow, underflow for Tg0,Tg1</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

Bits	Name	Memory Access	Description
1	RxClkCTrkOverflowUpdateTg0	R	<p>RxClkCTrkOverflowUpdateTg0: For RxClkDly tracking of tPHY_tDQS2DQ; overflow, underflow for Tg0,Tg1</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
0	RxClkCTrkUnderflowUpdateTg0	R	<p>RxClkCTrkUnderflowUpdateTg0: For RxClkDly tracking of tPHY_tDQS2DQ; overflow, underflow for Tg0,Tg1</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.91 PptInfoSel

- **Description:** PptInfoSel: Reserved for PHY training firmware use.
- **Size:** 10 bits
- **Offset:**  $(0x10000 + (j << 12)) + 0x9e$
- **Exists:** Always



**Table 13-95 Fields for Register: PptInfoSel**

Bits	Name	Memory Access	Description
9:0	PptInfoSel	R/W	PptInfoSel: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.92 Wck2DqoPptInfo

- **Description:** Wck2DqoPptInfo: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0x9f
- **Exists:** Always



**Table 13-96 Fields for Register: Wck2DqoPptInfo**

Bits	Name	Memory Access	Description
15:0	Wck2DqoPptInfo	R	<p>Wck2DqoPptInfo: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.93 Dqs2DqPptInfo

- **Description:** Dqs2DqPptInfo: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xa0
- **Exists:** Always



**Table 13-97 Fields for Register: Dqs2DqPptInfo**

Bits	Name	Memory Access	Description
15:0	Dqs2DqPptInfo	R	<p>Dqs2DqPptInfo: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.94 PptRxDqsTrackInfo

- **Description:** PptRxDqsTrackInfo: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xa1
- **Exists:** Always



**Table 13-98 Fields for Register: PptRxDqsTrackInfo**

Bits	Name	Memory Access	Description
15:0	PptRxDqsTrackInfo	R	<p>PptRxDqsTrackInfo: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.95 PptRxEnEvt

- Description:** PptRxEnEvt: This register is dynamically written by PHY Initialization Engine during frequency changes and should not be written by the user.
- Size:** 7 bits
- Offset:** (0x10000+(j<<12))+0xa2
- Exists:** Always



Table 13-99 Fields for Register: PptRxEnEvt

Bits	Name	Memory Access	Description
6:0	PptRxEnEvt	R/W	PptRxEnEvt: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.96 PptCtlStatic

- **Name:** Controls for the PPT of tDQS2DQ
- **Description:** PptCtlStatic: For LPDDR5/5X and LPDDR4X, Registers used for static configuration of the tDQS2DQ PPT hardware. Programmed by dwc\_ddrphy\_phyinit\_C\_initPhyConfig() to support frequency changes.
- **Size:** 12 bits
- **Offset:** (0x10000+(j<<12))+0xa3
- **Exists:** Always

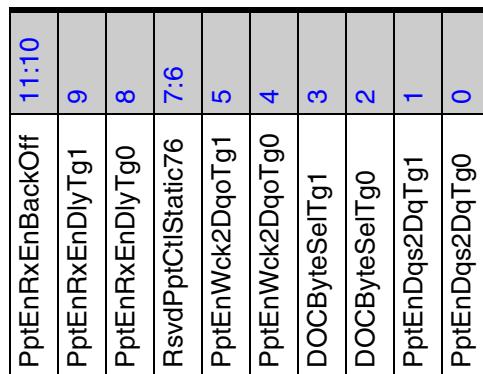


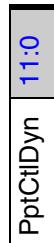
Table 13-100 Fields for Register: PptCtlStatic

Bits	Name	Memory Access	Description
11:10	PptEnRxEnBackOff	R/W	<p>PptEnRxEnBackOff: The number of UI to be subtracted from the delay position of the RxEnDly-trained dqs edge to position the RxEnDly in the middle of the preamble. This subtraction is performed by the DRAM drift compensation RxEn logic at the time PptRxEnEvnt is written to 0 when the DRAM drift compensation sequence is done, and the calculation is valid only when csrPptEnDqs2DqTg&lt;n&gt; is set.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
9	PptEnRxEnDlyTg1	R/W	<p>PptEnRxEnDlyTg1: Enables DRAM drift compensation correction to Register RxEnDlyTg1.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
8	PptEnRxEnDlyTg0	R/W	<p>PptEnRxEnDlyTg0: Enables DRAM drift compensation correction to Register RxEnDlyTg0.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
7:6	RsvdPptCtlStatic76	R/W	<p>RsvdPptCtlStatic76: Rsvd</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
5	PptEnWck2DqoTg1	R/W	<p>PptEnWck2DqoTg1: Enables DRAM drift compensation correction to Register RxClkDlyTg0.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
4	PptEnWck2DqoTg0	R/W	<p>PptEnWck2DqoTg0: Enables DRAM drift compensation correction to Register RxClkDlyTg0</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
3	DOCByteSelTg1	R/W	<p>DOCByteSelTg1: The DOCByteSelTg[0,1] allow flexibility in connecting the PHY DBYTEs to the DRAM DBYTEs. See JEDEC LPDDR4X sections 3.4, 3.14.19, 3.14.20, and 4.29. For each PHY DBYTE set to 0 if the DQS Oscillator Count will be read directly from the associated, directly connected DRAM dbyte; set to 1 if the DQS Oscillator Count will be sourced from the partner DBYTE. The MR18/MR19 DQS Oscillator Count is sourced by the DRAM LS byte (dq[7:0]). PHY DBYTE 0 and 1 are partners, 2 and 3 are partners, 4 and 5 are partners, 6 and 7 are partners, 8 and 9 are partners. Other DBYTE pairings are not supported.</p> <p><b>Note:</b> This configuration is per-timing-group. Having per-timing-group controls allows mixing of 16b and 8b DRAMs. If the DRAMs for TgX are only of the 8b type, then Register DOCByteSelTg[0,1] is set to 0, which; is the default value; other configurations require a non-default configuration of these Registers.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
2	DOCByteSelTg0	R/W	<p>DOCByteSelTg0: Controls for the PPT of tDQS2DQ</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
1	PptEnDqs2DqTg1	R/W	<p>PptEnDqs2DqTg1: Enables DRAM drift compensation correction to the timing group 1 (D1) Register TxDqDlyTg1.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
0	PptEnDqs2DqTg0	R/W	<p>PptEnDqs2DqTg0: Enables DRAM drift compensation correction to the timing group 0 (D0) Register TxDqDlyTg0.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.97 PptCtlDyn

- **Description:** PptCtlDyn: This register is dynamically written by PHY Initialization Engine during frequency changes and should not be written by the user.
- **Size:** 12 bits
- **Offset:**  $(0x10000 + (j << 12)) + 0xa4$
- **Exists:** Always



**Table 13-101 Fields for Register: PptCtlDyn**

Bits	Name	Memory Access	Description
11:0	PptCtlDyn	R/W	PptCtlDyn: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.98 RxTrainPattern8BitMode\_pX (for X = 0; X <= 3)

- **Description:** RxTrainPattern8BitMode\_pX: Reserved for PHY training firmware use.
- **Size:** 1 bit
- **Offset:** (0x10000+(j<<12))+0xa5+(X\*0x100000)
- **Exists:** Always



**Table 13-102 Fields for Register: RxTrainPattern8BitMode\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
0	RxTrainPattern8BitMode_p0	R/W	<p>RxTrainPattern8BitMode_p0: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.99 RxTrainPatternEnable

- **Description:** RxTrainPatternEnable: Reserved for PHY training firmware use.
- **Size:** 2 bits
- **Offset:**  $(0x10000 + (j \ll 12)) + 0xa6$
- **Exists:** Always



**Table 13-103 Fields for Register: RxTrainPatternEnable**

Bits	Name	Memory Access	Description
1:0	RxTrainPatternEnable	R/W	<p>RxTrainPatternEnable: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.100 TrainingParam

- **Description:** TrainingParam: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support LPDDR4X DRAM drift compensation.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xa7
- **Exists:** Always



**Table 13-104 Fields for Register: TrainingParam**

Bits	Name	Memory Access	Description
15:0	TrainingParam	R/W	<p>TrainingParam: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.101 DtsmByteCtrl0

- **Description:** DtsmByteCtrl0: Reserved for PHY training firmware use.
- **Size:** 7 bits
- **Offset:** (0x10000+(j<<12))+0xb0
- **Exists:** Always



**Table 13-105 Fields for Register: DtsmByteCtrl0**

Bits	Name	Memory Access	Description
6:0	DtsmByteCtrl0	R/W	<p>DtsmByteCtrl0: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.102 DtsmByteCtrl1

- **Description:** DtsmByteCtrl1: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() for protocols that support DRAM drift compensation.
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0xb1
- **Exists:** Always

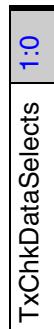


Table 13-106 Fields for Register: DtsmByteCtrl1

Bits	Name	Memory Access	Description
8:0	DtsmByteCtrl1	R/W	<p>DtsmByteCtrl1: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.103 TxChkDataSelects

- **Description:** TxChkDataSelects: Reserved for PHY training firmware use.
- **Size:** 2 bits
- **Offset:** (0x10000+(j<<12))+0xb2
- **Exists:** Always



**Table 13-107 Fields for Register: TxChkDataSelects**

Bits	Name	Memory Access	Description
1:0	TxChkDataSelects	R/W	TxChkDataSelects: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.104 DtsmGateInc

- **Description:** DtsmGateInc: Reserved for PHY training firmware use.
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0xb3
- **Exists:** Always



Table 13-108 Fields for Register: DtsmGateInc

Bits	Name	Memory Access	Description
8:0	DtsmGateInc	R/W	<p>DtsmGateInc: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.105 DtsmGateDec

- **Description:** DtsmGateDec: Reserved for PHY training firmware use.
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0xb4
- **Exists:** Always



**Table 13-109 Fields for Register: DtsmGateDec**

Bits	Name	Memory Access	Description
8:0	DtsmGateDec	R/W	<p>DtsmGateDec: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.106 DtsmLaneCtrl0\_iX (for X = 0; X <= 8)

- Description:** DtsmLaneCtrl0\_iX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support LPDDR4X DRAM drift compensation.
- Size:** 16 bits
- Offset:** (0x10000+(j<<12))+0xb5+(X\*0x100)
- Exists:** Always



Table 13-110 Fields for Register: DtsmLaneCtrl0\_iX (for X = 0; X <= 8)

Bits	Name	Memory Access	Description
15:0	DtsmLaneCtrl0_i0	R/W	<p>DtsmLaneCtrl0_i0: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.107 DtsmCmpCount\_iX (for X = 0; X <= 8)

- **Description:** DtsmCmpCount\_iX: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:**  $(0x10000 + (j \ll 12)) + 0xb6 + (X * 0x100)$
- **Exists:** Always



**Table 13-111 Fields for Register: DtsmCmpCount\_iX (for X = 0; X <= 8)**

Bits	Name	Memory Access	Description
15:0	DtsmCmpCount_i0	R	<p>DtsmCmpCount_i0: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.108 DtsmErrCount\_iX (for X = 0; X <= 8)

- **Description:** DtsmErrCount\_iX: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xb7+(X\*0x100)
- **Exists:** Always



**Table 13-112 Fields for Register: DtsmErrCount\_iX (for X = 0; X <= 8)**

Bits	Name	Memory Access	Description
15:0	DtsmErrCount_i0	R	<p>DtsmErrCount_i0: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.109 DtsmGoodCount\_iX (for X = 0; X <= 8)

- **Description:** DtsmGoodCount\_iX: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xb8+(X\*0x100)
- **Exists:** Always



**Table 13-113 Fields for Register: DtsmGoodCount\_iX (for X = 0; X <= 8)**

Bits	Name	Memory Access	Description
15:0	DtsmGoodCount_i0	R	<p>DtsmGoodCount_i0: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.110 DtsmGoodBar

- **Description:** DtsmGoodBar: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support LPDDR4X DRAM drift compensation.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xb9
- **Exists:** Always

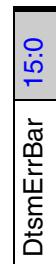


**Table 13-114 Fields for Register: DtsmGoodBar**

Bits	Name	Memory Access	Description
15:0	DtsmGoodBar	R/W	<p>DtsmGoodBar: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support LPDDR4X DRAM drift compensation.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.111 DtsmErrBar

- **Description:** DtsmErrBar: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support LPDDR4X DRAM drift compensation.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xba
- **Exists:** Always



**Table 13-115 Fields for Register: DtsmErrBar**

Bits	Name	Memory Access	Description
15:0	DtsmErrBar	R/W	<p>DtsmErrBar: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support LPDDR4X DRAM drift compensation.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.112 DtsmCountClears

- **Description:** DtsmCountClears: Reserved for PHY training firmware use.
- **Size:** 10 bits
- **Offset:**  $(0x10000 + (j << 12)) + 0xbb$
- **Exists:** Always



**Table 13-116 Fields for Register: DtsmCountClears**

Bits	Name	Memory Access	Description
9:0	DtsmCountClears	R/W	<p>DtsmCountClears: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.113 DtsmGoodThldXingInd

- **Description:** DtsmGoodThldXingInd: Reserved for PHY training firmware use.
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0xbc
- **Exists:** Always



**Table 13-117 Fields for Register: DtsmGoodThldXingInd**

Bits	Name	Memory Access	Description
8:0	DtsmGoodThldXingInd	R	<p>DtsmGoodThldXingInd: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.114 DtsmErrThldXingInd

- **Description:** DtsmErrThldXingInd: Reserved for PHY training firmware use.
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0xbd
- **Exists:** Always



**Table 13-118 Fields for Register: DtsmErrThldXingInd**

Bits	Name	Memory Access	Description
8:0	DtsmErrThldXingInd	R	<p>DtsmErrThldXingInd: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.115 TrainingIncDecDtsmEn\_rX (for X = 0; X <= 8)

- **Description:** TrainingIncDecDtsmEn\_rX: Reserved for PHY training firmware use.
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0xbe+(X\*0x100)
- **Exists:** Always

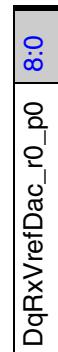


**Table 13-119 Fields for Register: TrainingIncDecDtsmEn\_rX (for X = 0; X <= 8)**

Bits	Name	Memory Access	Description
8:0	TrainingIncDecDtsmEn_r0	R/W	<p>TrainingIncDecDtsmEn_r0: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.116 DqRxVrefDac\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)

- **Description:** DqRxVrefDac\_rY\_pX: Rx VREF control for a TXRXDQ
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0xc8+(X\*0x100000)+(Y\*0x100)
- **Exists:** Always



**Table 13-120 Fields for Register: DqRxVrefDac\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 8)**

Bits	Name	Memory Access	Description
8:0	DqRxVrefDac_r0_p0	R/W	<p>DqRxVrefDac_r0_p0: Rx VREF control. Max allowed value is d'511. DAC voltage (V) = VDDQ (V) * DqRxVrefDac[8:0] / 512</p> <p><b>Value After Reset:</b> 0xff</p> <p><b>Exists:</b> Always</p>

### 13.1.117 RxReplicaLcdlPh1UI2UI

- **Description:** RxReplicaLcdlPh1UI2UI: For RxClkDly tracking of tPHY\_tDQS2DQ; RxReplica LCDL
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0xd5
- **Exists:** Always



**Table 13-121 Fields for Register: RxReplicaLcdlPh1UI2UI**

Bits	Name	Memory Access	Description
8:0	RxReplicaLcdlPh1UI2UI	R	<p>RxReplicaLcdlPh1UI2UI: If RxPubCalModels1UI=1, the phase control of the LCDL in the RxReplica cell that generates a 1UI delay.. If RxPubCalModels1UI=0, the phase control of the LCDL in the RxReplica cell that generates a 2UI delay.. Units of LCDL Phase stepsize.</p> <ul style="list-style-type: none"> <li>■ Value is used for calculating correction to the effective RxClkDly for changes in tPHY_tDQS2DQ.</li> <li>■ Value is captured with the assertion of csrRxReplicaStatusSnapNow.</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always  <b>Reset Mask:</b> 0x0  <b>Volatile:</b> true</p>

### 13.1.118 RxReplicaRatioNow

- **Description:** RxReplicaRatioNow: For RxClkDly tracking of tPHY\_tDQS2DQ; term used in correction.
- **Size:** 10 bits
- **Offset:** (0x10000+(j<<12))+0xd6
- **Exists:** Always



**Table 13-122 Fields for Register: RxReplicaRatioNow**

Bits	Name	Memory Access	Description
9:0	RxReplicaRatioNow	R	<p>RxReplicaRatioNow: The present value of (PathPhase divided by LcdlPh1UI). Units of csr*Dly LSB, nominally 1/64 UI in linear LCDL range. The value during training of RxClkDly should be read and used as the basis for writing RxReplicaRatioTrn which is used as baseline value in computing correction term to track changes in tPHY_tDQS2DQ. Value is captured with the assertion of csrRxReplicaStatusSnapNow.</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always  <b>Reset Mask:</b> 0x0  <b>Volatile:</b> true</p>

### 13.1.119 RxReplicaRxClkDlyCorrection

- **Name:** For RxClkDly tracking of tPHY\_tDQS2DQ; correction value.
- **Description:** RxReplicaRxClkDlyCorrection: Correction term applied to all the RxClkDly to get the effective RxClkDly to track VT changes in tPHY\_tDQS2DQ Value is captured with the assertion of csrRxReplicaStatusSnapNow.
- **Size:** 12 bits
- **Offset:** (0x10000+(j<<12))+0xd17
- **Exists:** Always

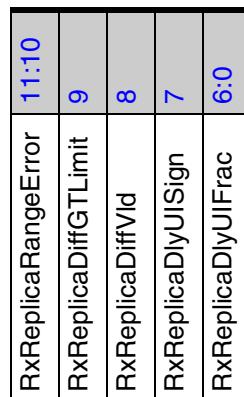


Table 13-123 Fields for Register: RxReplicaRxClkDlyCorrection

Bits	Name	Memory Access	Description
11:10	RxReplicaRangeError	R	<p>RxReplicaRangeError: If [10] is 1, the Initial Value of Phase Calibration is too high to complete properly. If [11] is 1, the Terminal Value of Phase Calibration is too low to complete properly. These bits are sticky and can only be cleared with via csrRxReplicaClearRangeError</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
9	RxReplicaDiffGTLimit	R	<p>RxReplicaDiffGTLimit: Set when the correction term applied to the RxClkDly to get the effective RxClkDly is computed to be greater than RxReplicaDiffMagMax</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

Bits	Name	Memory Access	Description
8	RxReplicaDiffVId	R	<p>RxReplicaDiffVId: Indicates that the snapshot of the 5 RxReplicaPathPhase[0..4], RxReplicaLcdlPh1UI,</p> <ul style="list-style-type: none"> <li>■ RxReplicaRatioNow, and RxReplicaRxClkDlyCorrection is complete and is a consistent set.</li> <li>■ Value is captured with the assertion of csrRxReplicaStatusSnapNow.</li> </ul> <p>Assumes that RxReplicaSelPathPhase[2:0] is static. If DiffVId=0, it may be that a new measurement and calculation are in-process. or that no measurement and calculation have been performed.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
7	RxReplicaDlyUISign	R	<p>RxReplicaDlyUISign: sign of correction</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
6:0	RxReplicaDlyUIFrac	R	<p>RxReplicaDlyUIFrac: magnitude of correction, units of fractional UI, nominally 1/64 UI in linear region.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.120 RxReplicaStatus00

- **Name:** For RxClkDly tracking of tPHY\_tDQS2DQ; valid signal for csr read values.
- **Description:** RxReplicaStatus00: RxReplica tPHY\_tDQS2DQ tracking information
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xd8
- **Exists:** Always



Table 13-124 Fields for Register: RxReplicaStatus00

Bits	Name	Memory Access	Description
15:8	RxRepDlyMaxPos	R	<p>RxRepDlyMaxPos: sign=0,magnitude[6:0] of the largest-magnitude positive correction since the last assertion of RxReplicaDlyRstMaxMin. (for debug)</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
7:0	RxRepDlyMinNeg	R	<p>RxRepDlyMinNeg: sign=1,magnitude[6:0] of the largest-magnitude negative correction since the last assertion of RxReplicaDlyRstMaxMin. (for debug)</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.121 RxReplicaUICalWait

- **Description:** RxReplicaUICalWait: RxReplica UI calibration wait time
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xd9
- **Exists:** Always



**Table 13-125 Fields for Register: RxReplicaUICalWait**

Bits	Name	Memory Access	Description
15:0	RxReplicaUICalWait	R/W	<p>RxReplicaUICalWait: Counter that tells how many DfiClks to wait for UI calibration (of RxReplica's LCDL) to run</p> <p><b>Value After Reset:</b> 0x80</p> <p><b>Exists:</b> Always</p>

### 13.1.122 RxReplicaInterval

- **Description:** RxReplicaInterval: RxReplica calibration interval time
- **Size:** 16 bits
- **Offset:**  $(0x10000 + (j << 12)) + 0xda$
- **Exists:** Always



**Table 13-126 Fields for Register: RxReplicaInterval**

Bits	Name	Memory Access	Description
15:0	RxReplicaInterval	R/W	<p>RxReplicaInterval: Counter that tells how many DfiClks to wait for next RxReplica complete calibration (phase + UI) after a successful calibration</p> <p><b>Value After Reset:</b> 0x1200</p> <p><b>Exists:</b> Always</p>

### 13.1.123 RxReplicaDontGateRCTMath

- **Description:** RxReplicaDontGateRCTMath: Keep this zero. If 1, disables the power saving logic for RCT Math
- **Size:** 1 bit
- **Offset:** (0x10000+(j<<12))+0xdb
- **Exists:** Always



**Table 13-127 Fields for Register: RxReplicaDontGateRCTMath**

Bits	Name	Memory Access	Description
0	RxReplicaDontGateRCTMath	R/W	<p>RxReplicaDontGateRCTMath: Keep this zero. If 1, disables the power saving logic for RCT Math</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.124 DbyteCntrl

- **Name:** Misc Control CSR for DBYTE
- **Description:** DbyteCntrl: Controls CSR related to DBYTE
- **Size:** 2 bits
- **Offset:** (0x10000+(j<<12))+0xde
- **Exists:** Always



**Table 13-128 Fields for Register: DbyteCntrl**

Bits	Name	Memory Access	Description
1	LoopBackDisWckTri	R/W	<p>LoopBackDisWckTri: [Enable only when LoopBackEn ==1] Ensures that WCK will not be tristated after first WCK transaction.</p> <p><b>Value After Reset:</b> 0x0 <b>Exists:</b> Always</p>
0	LoopBackDisDqsTri	R/W	<p>LoopBackDisDqsTri: [Enable only when LoopBackEn ==1] Ensures that DQS will not be tristated after first write transaction.</p> <p><b>Value After Reset:</b> 0x0 <b>Exists:</b> Always</p>

### 13.1.125 DbCurrentDlyTimingInfoTgX (for X = 0; X <= 1)

- **Description:** DbCurrentDlyTimingInfoTgX: Reserved for PHY training firmware use and for debugging
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe4+(X\*0x1)
- **Exists:** Always



**Table 13-129 Fields for Register: DbCurrentDlyTimingInfoTgX (for X = 0; X <= 1)**

Bits	Name	Memory Access	Description
15:0	DbCurrentDlyTimingInfoTg0	R	<p>DbCurrentDlyTimingInfoTg0: Reserved for PHY training firmware use and for debugging</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.126 DxRxPowerDown

- **Name:** Dq/Dqs receiver control
- **Description:** DxRxPowerDown: Misc control of rxdq cell and rxdqs cell
- **Size:** 1 bit
- **Offset:** (0x10000+(j<<12))+0xfa
- **Exists:** Always

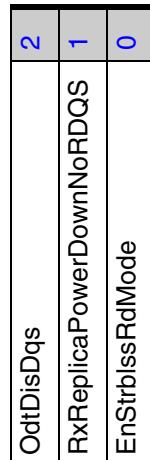


**Table 13-130 Fields for Register: DxRxPowerDown**

Bits	Name	Memory Access	Description
0	RxReplicaPowerDown	R/W	<p>RxReplicaPowerDown: Active high signal which powers down the receiver in the RxReplica CKT in the DQS DIFF Slice. After this pin is deasserted the receiver cannot be used for a minimum of 100 ns. See RxReplicaPowerDownNoRDQS, a per p-state control used in strobeless read mode, ORed with this control.</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

### 13.1.127 RxDigStrbEn\_pX (for X = 0; X <= 3)

- **Name:** Enable Digital Strobe Read Mode.
- **Description:** RxDigStrbEn\_pX: Controls enable and power savings for strobeless read mode, per p-state.
- **Size:** 3 bits
- **Offset:** (0x10000+(j<<12))+0xfb+(X\*0x10000)
- **Exists:** Always



**Table 13-131 Fields for Register: RxDigStrbEn\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
2	OdtDisDqs	R/W	<p>OdtDisDqs: When asserted csrOdtDisDqs disables DQS DIFF ODT per pstate; That is it overrides csr OdtEnDqs. Default to 0, nonactive.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
1	RxReplicaPowerDownNoRDQS	R/W	<p>RxReplicaPowerDownNoRDQS: Active high signal which powers down the RxReplica differential receiver circuit. ORed with the non-pstated RxReplicaPowerDown. Default to 0, nonactive.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
0	EnStrbLssRdMode	R/W	<p>EnStrbLssRdMode: Enables LPDDR5 strobeless mode. Must be 0 if LPDDR5RdqEn=1. In this mode RxDigStrbDly is used to control timing of RDQ/RDBI sampling from the hard macro SE pclk. In this mode RDQS DQS_T,DQS_C and RxClkT2UIDly[6:0], RxClkC2UIDly[6:0] are not used. In this mode, RxClkT2UIDly[9:7] are used to select the read-data-fifo RxPtr. CsrDxDigStrobeMode needs to be programmed with a value of 2'10. CsrRxPubCalModels1UI must be configured with the same value as register HMRxLcdlSeed csr RxCalModels1UI. See JEDEC Spec. MR20[1:0] needs to be programmed accordingly. For power savings in this mode, set csrDfiClkDqsDis For power savings in this mode, set csrPClkDqsDis For power savings in this mode, set csrRxReplicaPowerDownNoRDQS For power savings in this mode, set csrOdtEnDqs</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.128 DxPipeEn\_pX (for X = 0; X <= 3)

- **Description:** DxPipeEn\_pX: Deprecated
- **Size:** 2 bits
- **Offset:** (0x10000+(j<<12))+0xfc+(X\*0x100000)
- **Exists:** Always



**Table 13-132 Fields for Register: DxPipeEn\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
1	DxRdPipeEn	R/W	DxRdPipeEn: Deprecated <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
0	DxWrPipeEn	R/W	DxWrPipeEn: Deprecated <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.129 LcdlCalCtrl

- **Description:** LcdlCalCtrl: Reserved for PHY test firmware use.
- **Size:** 6 bits
- **Offset:** (0x10000+(j<<12))+0xfe
- **Exists:** Always



**Table 13-133 Fields for Register: LcdlCalCtrl**

Bits	Name	Memory Access	Description
5:0	LcdlCalCtrl	R/W	LcdlCalCtrl: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.130 PclkDCDCtrl\_pX (DBYTE) (for X = 0; X <= 3)

- **Description:** PclkDCDCtrl\_pX: Controls the DCD Comparator for DCA calibration
- **Size:** 2 bits
- **Offset:** (0x10000+(j<<12))+0x100+(X\*0x100000)
- **Exists:** Always



**Table 13-134 Fields for Register: PclkDCDCtrl\_pX (DBYTE) (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
1	PclkDCDOffsetMode	R/W	PclkDCDOffsetMode: Puts the comparator in offset calibration mode <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
0	PclkDCDEn	R/W	PclkDCDEn: Enables the comparator <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.131 PPTTrainSetup2\_pX (for X = 0; X <= 3)

- **Description:** PPTTrainSetup2\_pX: Programmed by PHY training firmware to support LPDDR4X DRAM drift compensation.
- **Size:** 11 bits
- **Offset:** (0x10000+(j<<12))+0x102+(X\*0x100000)
- **Exists:** Always



Table 13-135 Fields for Register: PPTTrainSetup2\_pX (for X = 0; X <= 3)

Bits	Name	Memory Access	Description
10:0	PPTTrainSetup2_p0	R/W	PPTTrainSetup2_p0: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.132 ForceInternalUpdate

- **Description:** ForceInternalUpdate: This Register used by Training Firmware to force an internal PHY Update Event.
- **Size:** 1 bit
- **Offset:** (0x10000+(j<<12))+0x103
- **Exists:** Always



**Table 13-136 Fields for Register: ForceInternalUpdate**

Bits	Name	Memory Access	Description
0	ForceInternalUpdate	R/W	<p>ForceInternalUpdate: This Register is used by Training Firmware to force an internal PHY Update Event. Optionally can be used by System Software to issue a PHY update Event. The register may be written to 1'b1 only when all of the following are true:</p> <ul style="list-style-type: none"> <li>■ The DFI interface is offline.</li> <li>■ The Training Hardware is sending only DES and all transactions have retired.</li> </ul> <p>Additionally, this Register must stay asserted for at least 32 DFICLKs before clearing. Prematurely clearing this register may result in an incomplete update.</p> <p><b>Value After Reset:</b> 0x0 <b>Exists:</b> Always</p>

### 13.1.133 ForceRxDataFifoUpd

- **Description:** ForceRxDataFifoUpd: This register used to force update event in RxDataFifo
- **Size:** 1 bit
- **Offset:** (0x10000+(j<<12))+0x104
- **Exists:** Always



**Table 13-137 Fields for Register: ForceRxDataFifoUpd**

Bits	Name	Memory Access	Description
0	ForceRxDataFifoUpd	R/W	<p>ForceRxDataFifoUpd: Used for internal diagnostic testing</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.134 DMIPinPresent\_pX (for X = 0; X <= 3)

- **Name:** This Register is used to enable the Read-DBI function in each DBYTE
- **Description:** DMIPinPresent\_pX: This Register is used to instruct the PHY whether to enable the Read-DBI function on Lane8 of each DBYTE. LPDDR4X and LPDDR5 Protocols support the Read-DBI feature.
- **Size:** 1 bit
- **Offset:** (0x10000+(j<<12))+0x108+(X\*0x100000)
- **Exists:** Always



**Table 13-138 Fields for Register: DMIPinPresent\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
0	RdDbiEnabled	R/W	<p>RdDbiEnabled: This bit must be set to 1'b1 if Read-DBI is enabled in a connected LPDDR4X/5 device or RD ECC is enabled in LPDDR5 device.</p> <ul style="list-style-type: none"> <li>■ If set, the following DRAM MR should also be set [LPDDR4X/5.MR3.OP[7]=1] for Read DBI.</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.135 InhibitTxRdPtrInit\_pX (for X = 0; X <= 3)

- **Description:** InhibitTxRdPtrInit\_pX: For use by Training FW
- **Size:** 2 bits
- **Offset:** (0x10000+(j<<12))+0x10b+(X\*0x100000)
- **Exists:** Always



**Table 13-139 Fields for Register: InhibitTxRdPtrInit\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
1:0	InhibitTxRdPtrInit_p0	R/W	<p>InhibitTxRdPtrInit_p0: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.136 AllowInhibitTxRdPtrInit

- **Description:** AllowInhibitTxRdPtrInit: For use by Training FW
- **Size:** 2 bits
- **Offset:** (0x10000+(j<<12))+0x10c
- **Exists:** Always



Table 13-140 Fields for Register: AllowInhibitTxRdPtrInit

Bits	Name	Memory Access	Description
1:0	AllowInhibitTxRdPtrInit	R/W	<b>AllowInhibitTxRdPtrInit:</b> Reserved for Synopsys internal use <b>Value After Reset:</b> 0x3 <b>Exists:</b> Always

### 13.1.137 RdfPtrChkStatusWptrT

- **Description:** RdfPtrChkStatusWptrT: Error status for RDF WptrT checker
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0x15d
- **Exists:** Always

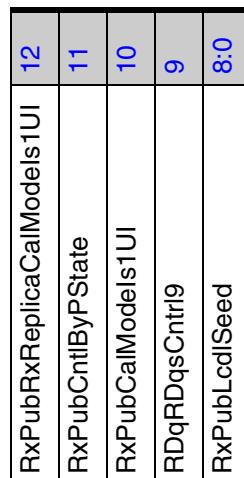


**Table 13-141 Fields for Register: RdfPtrChkStatusWptrT**

Bits	Name	Memory Access	Description
8:0	RdfPtrChkStatusWptrT	R	<p>RdfPtrChkStatusWptrT: RDF pointer checker WptrT error, per-lane.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.138 RDqRDqsCntrl\_pX (for X = 0; X <= 3)

- **Name:** Dq/Dqs receiver control
- **Description:** RDqRDqsCntrl\_pX: Controls to support rx pipe generation of hardmacro signals having analog timing requirements. Used for LP5 read using digital strobe mode; RxDigStrbEn EnStrbLssRdMode=1, for both RxCalModeIs1UI=0 and RxCalModeIs1UI=1.
- **Size:** 13 bits
- **Offset:** (0x10000+(j<<12))+0x15f+(X\*0x100000)
- **Exists:** Always



**Table 13-142 Fields for Register: RDqRDqsCntrl\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
12	RxPubRxReplicaCalModels1UI	R/W	RxPubRxReplicaCalModels1UI: A PUB version of the hardmacro control csr RxReplicaCalModels1UI that must be configured the same as that csr. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
11	RxPubCntrlByPState	R/W	RxPubCntrlByPState: reserved. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
10	RxPubCalModels1UI	R/W	RxPubCalModels1UI: Used to interpret the RxPubLcdlSeed scaling Used by the pub rx pipe and not used by any LCDL nor delay scaling. Suggested configuration is the configuration of HMDBYTE RxCalModels1UI In LP5 mode, used to support RxDigStrbDly timing control <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

Bits	Name	Memory Access	Description
9	RDqRDqsCntrl9	R/W	<p>RDqRDqsCntrl9: Reserved</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
8:0	RxPubLcdlSeed	R/W	<p>RxPubLcdlSeed: Used by the pub rx pipe and not used by any LCDL. used to support RxStandby (csrDxRxStandbyEn) power savings. The most number of pclk periods for exit and entry will be used for the fastest DataRate. for which the RxPubLcdlSeed[8:4] is less than 0x0F. The lower 4b of RxLcdlSeed are not needed here. For best power savings, RxPubLcdlSeed[ should be configured with the trained value HMDBYTE RxLcdlSeed[8:j0]. The default, reset value of zero is safe at all frequencies but will allow greater power in the rx receivers than may be necessary.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.139 RxReplicaLcdlCalCtrl

- **Description:** RxReplicaLcdlCalCtrl: Reserved for PHY test firmware use.
- **Size:** 8 bits
- **Offset:** (0x10000+(j<<12))+0x189
- **Exists:** Always



**Table 13-143 Fields for Register: RxReplicaLcdlCalCtrl**

Bits	Name	Memory Access	Description
7:0	RxReplicaLcdlCalCtrl	R/W	RxReplicaLcdlCalCtrl: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.140 RxReplicaRangeVal\_pX (for X = 0; X <= 3)

- **Description:** RxReplicaRangeVal\_pX: Controls for RxClk timing modes.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0x209+(X\*0x100000)
- **Exists:** Always



**Table 13-144 Fields for Register: RxReplicaRangeVal\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:8	RxReplicaShortCalRangeB	R/W	<p>RxReplicaShortCalRangeB: This field determines the phase range we need to scan on both sides of the selected calibration phase This value is selected when the csr ShortCalRangeSel is set to 1</p> <p><b>Value After Reset:</b> 0x4</p> <p><b>Exists:</b> Always</p>
7:0	RxReplicaShortCalRangeA	R/W	<p>RxReplicaShortCalRangeA: This field determines the phase range we need to scan on both sides of the selected calibration phase This value is selected when the csr ShortCalRangeSel is set to 0</p> <p><b>Value After Reset:</b> 0x4</p> <p><b>Exists:</b> Always</p>

### 13.1.141 RxReplicaCtl04\_pX (for X = 0; X <= 3)

- **Name:** For RxClkDly tracking of tPHY\_tDQS2DQ; control for RxReplica
- **Description:** RxReplicaCtl04\_pX: Configuration Parameter for computing RxReplicaRxClkDlyCorrection
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0x20f+(X\*0x100000)
- **Exists:** Always

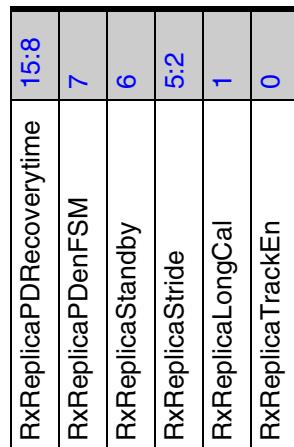


Table 13-145 Fields for Register: RxReplicaCtl04\_pX (for X = 0; X <= 3)

Bits	Name	Memory Access	Description
15:8	RxReplicaPDRecoverytime	R/W	RxReplicaPDRecoverytime: Programmed in number of DfiClks for RxReplica Receiver's Powerdown exit Recovery time. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	RxReplicaPDenFSM	R/W	RxReplicaPDenFSM: When =1, PowerDown pin of RxReplica receiver is driven by RxRepCtrl FSM in the PUB. When =0, (RxReplicaPowerDown   RxReplicaPowerDownNoRDQS) controls the PowerDown pin for RxReplica receiver. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6	RxReplicaStandby	R/W	RxReplicaStandby: This field forces Standby pin of RxReplica receiver to low. Used only for Debug purposes. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
5:2	RxReplicaStride	R/W	RxReplicaStride: This field determines the phase count size to increment in the path phase calibration experiment. <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always

Bits	Name	Memory Access	Description
1	RxReplicaLongCal	R/W	<p>RxReplicaLongCal: This bit determines whether we want a longer or shorter calibration. (It should be set to one for the initial calibration)</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>
0	RxReplicaTrackEn	R/W	<p>RxReplicaTrackEn: Set this bit zero to save power when we are not running RxReplica tracking. Enabled by default</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

### 13.1.142 RdfPtrChkStatusWptrC

- **Description:** RdfPtrChkStatusWptrC: Error status for RDF WptrT checker
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0x25d
- **Exists:** Always



**Table 13-146 Fields for Register: RdfPtrChkStatusWptrC**

Bits	Name	Memory Access	Description
8:0	RdfPtrChkStatusWptrC	R	<p>RdfPtrChkStatusWptrC: RDF pointer checker WptrT error, per-lane.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.143 RxReplicaPathPhase0\_pX (for X = 0; X <= 3)

- **Description:** RxReplicaPathPhase0\_pX: For RxClkDly tracking of tPHY\_tDQS2DQ; RxReplica path delay
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0x2a0+(X\*0x100000)
- **Exists:** Always



**Table 13-147 Fields for Register: RxReplicaPathPhase0\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
8:0	RxReplicaPathPhase0_p0	R/W	<p>RxReplicaPathPhase0_p0: Measurement result by RxReplica cell of a replica of the path from DQS pad to RxClk LCDL. Units of LCDL Phase stepsize. Smallest value of RxReplicaPhase which causes the RxReplica phd_out[0] to set</p> <ul style="list-style-type: none"> <li>■ Value will be used for correcting the effective RxClkDly for changes in tPHY_tDQS2DQ</li> <li>■ Value is snapshot of the present value, of the pstate being run.</li> <li>■ Value is captured with the assertion of csrRxReplicaStatusSnapNow.</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.1.144 RxReplicaPathPhase1\_pX (for X = 0; X <= 3)

- **Description:** RxReplicaPathPhase1\_pX: For RxClkDly tracking of tPHY\_tDQS2DQ; RxReplica path delay
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0x2a1+(X\*0x100000)
- **Exists:** Always



**Table 13-148 Fields for Register: RxReplicaPathPhase1\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
8:0	RxReplicaPathPhase1_p0	R/W	<p>RxReplicaPathPhase1_p0: Measurement result by RxReplica cell of a replica of the path from DQS pad to RxClk LCDL. Units of LCDL Phase stepsize. Smallest value of RxReplicaPhase which causes the RxReplica phd_out[1] to set</p> <ul style="list-style-type: none"> <li>■ Value will be used for correcting the effective RxClkDly for changes in tPHY_tDQS2DQ</li> <li>■ Value is snapshot of the present value, of the pstate being run.</li> <li>■ Value is captured with the assertion of csrRxReplicaStatusSnapNow.</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.1.145 RxReplicaPathPhase2\_pX (for X = 0; X <= 3)

- **Description:** RxReplicaPathPhase2\_pX: For RxClkDly tracking of tPHY\_tDQS2DQ; RxReplica path delay
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0x2a2+(X\*0x100000)
- **Exists:** Always



**Table 13-149 Fields for Register: RxReplicaPathPhase2\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
8:0	RxReplicaPathPhase2_p0	R/W	<p>RxReplicaPathPhase2_p0: Measurement result by RxReplica cell of a replica of the path from DQS pad to RxClk LCDL. Units of LCDL Phase stepsize. Smallest value of RxReplicaPhase which causes the RxReplica phd_out[2] to set</p> <ul style="list-style-type: none"> <li>■ Value will be used for correcting the effective RxClkDly for changes in tPHY_tDQS2DQ</li> <li>■ Value is snapshot of the present value, of the pstate being run.</li> <li>■ Value is captured with the assertion of csrRxReplicaStatusSnapNow.</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.1.146 RxReplicaPathPhase3\_pX (for X = 0; X <= 3)

- **Description:** RxReplicaPathPhase3\_pX: For RxClkDly tracking of tPHY\_tDQS2DQ; RxReplica path delay
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0x2a3+(X\*0x100000)
- **Exists:** Always



**Table 13-150 Fields for Register: RxReplicaPathPhase3\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
8:0	RxReplicaPathPhase3_p0	R/W	<p>RxReplicaPathPhase3_p0: Measurement result by RxReplica cell of a replica of the path from DQS pad to RxClk LCDL. Units of LCDL Phase stepsize. Smallest value of RxReplicaPhase which causes the RxReplica phd_out[3] to set</p> <ul style="list-style-type: none"> <li>■ Value will be used for correcting the effective RxClkDly for changes in tPHY_tDQS2DQ</li> <li>■ Value is snapshot of the present value, of the pstate being run.</li> <li>■ Value is captured with the assertion of csrRxReplicaStatusSnapNow.</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.1.147 RxReplicaPathPhase4\_pX (for X = 0; X <= 3)

- **Description:** RxReplicaPathPhase4\_pX: For RxClkDly tracking of tPHY\_tDQS2DQ; RxReplica path delay
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0x2a4+(X\*0x100000)
- **Exists:** Always

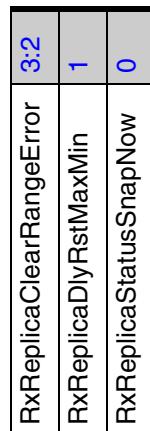


**Table 13-151 Fields for Register: RxReplicaPathPhase4\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
8:0	RxReplicaPathPhase4_p0	R/W	<p>RxReplicaPathPhase4_p0: Measurement result by RxReplica cell of a replica of the path from DQS pad to RxClk LCDL. Units of LCDL Phase stepsize. Smallest value of RxReplicaPhase which causes the RxReplica phd_out[4] to set</p> <ul style="list-style-type: none"> <li>■ Value will be used for correcting the effective RxClkDly for changes in tPHY_tDQS2DQ</li> <li>■ Value is snapshot of the present value, of the pstate being run.</li> <li>■ Value is captured with the assertion of csrRxReplicaStatusSnapNow.</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.1.148 RxReplicaCtl00

- **Name:** For RxClkDly tracking of tPHY\_tDQS2DQ; for reading status
- **Description:** RxReplicaCtl00: Control for reading RxReplica tPHY\_tDQS2DQ tracking information
- **Size:** 4 bits
- **Offset:** (0x10000+(j<<12))+0x2ac
- **Exists:** Always



**Table 13-152 Fields for Register: RxReplicaCtl00**

Bits	Name	Memory Access	Description
3:2	RxReplicaClearRangeError	R/W	RxReplicaClearRangeError: Used to clear the sticky RxReplicaRangeError bits in csrRxReplicaRxClkDlyCorrection <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
1	RxReplicaDlyRstMaxMin	R/W	RxReplicaDlyRstMaxMin: When set it resets, i.e., zeroes, RxRepDlyMinNeg and RxRepDlyMaxPos <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
0	RxReplicaStatusSnapNow	R/W	RxReplicaStatusSnapNow: On assertion of this csr (i.e., the leading edge), the values of the 5 RxReplicaPathPhase[0..4], RxReplicaLcdlPh1UI, RxReplicaRxClkDlyCorrection, <ul style="list-style-type: none"> <li>■ RxReplicaRatioNow, and the status bits RxReplicaDiffVld and RxReplicaDiffGTLimit are latched for reading.</li> </ul> <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.149 RxReplicaCtl01\_pX (for X = 0; X <= 3)

- **Name:** For RxClkDly tracking of tPHY\_tDQS2DQ; control for RxReplica
- **Description:** RxReplicaCtl01\_pX: Configuration Parameter for computing RxReplicaRxClkDlyCorrection
- **Size:** 3 bits
- **Offset:**  $(0x10000 + (j \ll 12)) + 0x2ad + (X * 0x100000)$
- **Exists:** Always



**Table 13-153 Fields for Register: RxReplicaCtl01\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
2:0	RxReplicaSelPathPhase	R/W	<p>RxReplicaSelPathPhase: Specify which RxReplicaPathPhase[0..4] to use for computing RxReplicaRatioNow. If set to greater than 4, a value of zero is assumed.</p> <p><b>Value After Reset:</b> 0x2</p> <p><b>Exists:</b> Always</p>

### 13.1.150 RxReplicaCtl02\_pX (for X = 0; X <= 3)

- **Name:** For RxClkDly tracking of tPHY\_tDQS2DQ; correction limit.
- **Description:** RxReplicaCtl02\_pX: Parameter used in computing RxReplicaRxClkDlyCorrection
- **Size:** 7 bits
- **Offset:** (0x10000+(j<<12))+0x2ae+(X\*0x100000)
- **Exists:** Always



**Table 13-154 Fields for Register: RxReplicaCtl02\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
6:0	RxReplicaDiffLimit	R/W	<p>RxReplicaDiffLimit: Maximum correction term applied to the RxClkDly to get the effective RxClkDly to track variations in tPHY_tDQS2DQ. Units of csr*Dly LSB, nominally 1/64 UI in linear LCDL range. Suggested value is one-half UI minus 1; per p-state or 31=32-1 when operating in a frequency range for which the LCDL is linear.</p> <p><b>Value After Reset:</b> 0x1f</p> <p><b>Exists:</b> Always</p>

### 13.1.151 RxReplicaCtl03\_pX (for X = 0; X <= 3)

- **Name:** For RxClkDly tracking of tPHY\_tDQS2DQ; Baseline value.
- **Description:** RxReplicaCtl03\_pX: Baseline value used in computing RxReplicaRxClkDlyCorrection
- **Size:** 10 bits
- **Offset:** (0x10000+(j<<12))+0x2af+(X\*0x100000)
- **Exists:** Always



**Table 13-155 Fields for Register: RxReplicaCtl03\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
9:0	RxReplicaRatioTrn	R/W	<p>RxReplicaRatioTrn: Normally set to RxReplicaRatioNow at the time of training RxClkDly. Units of csr*Dly LSB, nominally 1/64 UI in linear LCDL range. Used as baseline value in computing correction term to track changes in tPHY_tDQS2DQ.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.152 RdfPtrChkStatusRptr

- **Description:** RdfPtrChkStatusRptr: Error status for RDF WptrT checker
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0x35d
- **Exists:** Always

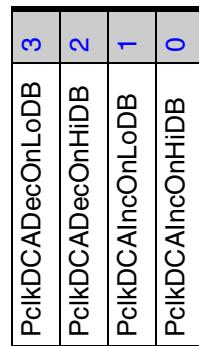


**Table 13-156 Fields for Register: RdfPtrChkStatusRptr**

Bits	Name	Memory Access	Description
8:0	RdfPtrChkStatusRptr	R	<p>RdfPtrChkStatusRptr: RDF pointer checker WptrT error, per-lane.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.153 PclkDCACalCtrl0DB

- **Name:** Controls DCA sample polarity
- **Description:** PclkDCACalCtrl0DB: Controls DCA sample polarity. Used by PHYINIT.
- **Size:** 4 bits
- **Offset:** (0x10000+(j<<12))+0x800
- **Exists:** Always



**Table 13-157 Fields for Register: PclkDCACalCtrl0DB**

Bits	Name	Memory Access	Description
3	PclkDCADecOnLoDB	R/W	PclkDCADecOnLoDB: Decrement DCA value on low samples <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always
2	PclkDCADecOnHiDB	R/W	PclkDCADecOnHiDB: Decrement DCA value on high samples <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
1	PclkDCAIncOnLoDB	R/W	PclkDCAIncOnLoDB: Increment DCA value on low samples <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
0	PclkDCAIncOnHiDB	R/W	PclkDCAIncOnHiDB: Increment DCA value on high samples <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always

### 13.1.154 PclkDCADynCtrl

- **Name:** Dynamic bits for DCA control
- **Description:** PclkDCADynCtrl: Dynamic bits for DCA control. Used by PHYINIT.
- **Size:** 4 bits
- **Offset:** (0x10000+(j<<12))+0x802
- **Exists:** Always

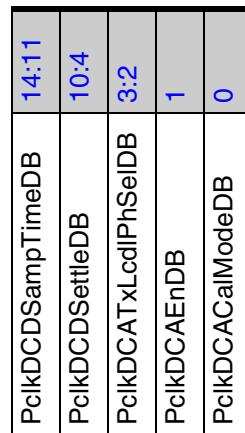


**Table 13-158 Fields for Register: PclkDCADynCtrl**

Bits	Name	Memory Access	Description
3	PclkDCAForceUpd	R/W	PclkDCAForceUpd: Controls code update pin of PCLK RX DCA macro when PclkCalReset=1 <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
2	PclkDCAForceSampVld	R/W	PclkDCAForceSampVld: Forces CalSeq to take a manual sample <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
1	PclkDCAQuickSearch	R/W	PclkDCAQuickSearch: 1 = Do quick search. 0 = Do full search. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
0	PclkDCACalReset	R/W	PclkDCACalReset: De-assert this to begin DCA calibration Must be kept de-asserted until PclkDcaDone=1 <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always

### 13.1.155 PclkDCAStaticCtrl0DB\_pX (for X = 0; X <= 3)

- **Name:** Static bits for DCA control in DBYTE blocks
- **Description:** PclkDCAStaticCtrl0DB\_pX: Static bits for DCA control in DBYTE blocks. Used by PHYINIT.
- **Size:** 15 bits
- **Offset:** (0x10000+(j<<12))+0x803+(X\*0x100000)
- **Exists:** Always



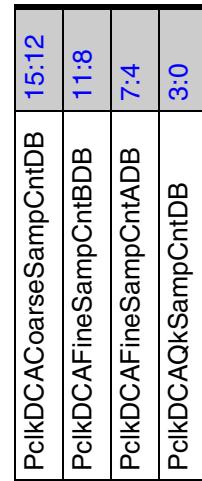
**Table 13-159 Fields for Register: PclkDCAStaticCtrl0DB\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
14:11	PclkDCDSampTimeDB	R/W	<p>PclkDCDSampTimeDB: Set by PHYINIT when PCLK DCA is using Comparator Indicates how long to delay after SampCkEn is issued Delay will be (VALUE) DFICLKs. Reset VALUE is 2.</p> <p><b>Value After Reset:</b> 0x2</p> <p><b>Exists:</b> Always</p>
10:4	PclkDCDSettleDB	R/W	<p>PclkDCDSettleDB: Set by PHYINIT when PCLK DCA is using Comparator Indicates how long to delay after RXDCA code update pulse is issued Delay will [(2*VALUE)-1] DFICLKs. Reset VALUE is 4.</p> <p><b>Value After Reset:</b> 0x4</p> <p><b>Exists:</b> Always</p>
3:2	PclkDCATxLcdlPhSelDB	R/W	<p>PclkDCATxLcdlPhSelDB: Selects the DlySteps input (aka Phase) to lcdl_tx 00 = Use TxClk_Phase_Cal 01 = Use (TxClk_Phase_Cal/2) 10 = Use csrPclkDCATxLcdlPhase 11 = Use value chosen by TxTimingSel</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
1	PclkDCAEnDB	R/W	PclkDCAEnDB: 1 = Enable DCA if PClkEn=1. 0 = Disable DCA <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
0	PclkDCACalModeDB	R/W	PclkDCACalModeDB: 1 = Use LCDL for PhDet 0 = Use DCD for PhDet <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.156 PclkDCASampCntDB

- **Name:** Sample counts for various stages of DCA calibration
- **Description:** PclkDCASampCntDB: Sample counts for various stages of DCA calibration. Used by PHYINIT.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0x804
- **Exists:** Always



**Table 13-160 Fields for Register: PclkDCASampCntDB**

Bits	Name	Memory Access	Description
15:12	PclkDCACoarseSampCntDB	R/W	PclkDCACoarseSampCntDB: Number of coarse samples for full search <b>Value After Reset:</b> 0x4 <b>Exists:</b> Always
11:8	PclkDCAFineSampCntBDB	R/W	PclkDCAFineSampCntBDB: Number of fine samples for full search in UR or LL <b>Value After Reset:</b> 0x6 <b>Exists:</b> Always
7:4	PclkDCAFineSampCntADB	R/W	PclkDCAFineSampCntADB: Number of fine samples for full search on midline <b>Value After Reset:</b> 0x6 <b>Exists:</b> Always
3:0	PclkDCAQkSampCntDB	R/W	PclkDCAQkSampCntDB: Number of samples for quick search <b>Value After Reset:</b> 0x3 <b>Exists:</b> Always

### 13.1.157 PclkDCAHysMaskDB

- **Description:** PclkDCAHysMaskDB: Used by DCA quick search algorithm
- **Size:** 3 bits
- **Offset:**  $(0x10000 + (j << 12)) + 0x805$
- **Exists:** Always



**Table 13-161 Fields for Register: PclkDCAHysMaskDB**

Bits	Name	Memory Access	Description
2:0	PclkDCAHysMaskDB	R/W	PclkDCAHysMaskDB: Set one bit for every quick search sample. Must always be 0x7. <b>Value After Reset:</b> 0x7 <b>Exists:</b> Always

### 13.1.158 PclkDCACalFineBoundDB

- **Name:** Sets limit values of DCA Fine searches
- **Description:** PclkDCACalFineBoundDB: Sets limit values of DCA Fine searches. Used by PHYINIT.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0x806
- **Exists:** Always

PclkDCALLMinFineDB	15:12
PclkDCALLMaxFineDB	11:8
PclkDCAURMinFineDB	7:4
PclkDCAURMaxFineDB	3:0

Table 13-162 Fields for Register: PclkDCACalFineBoundDB

Bits	Name	Memory Access	Description
15:12	PclkDCALLMinFineDB	R/W	PclkDCALLMinFineDB: LL Min Fine <b>Value After Reset:</b> 0xe <b>Exists:</b> Always
11:8	PclkDCALLMaxFineDB	R/W	PclkDCALLMaxFineDB: LL Max Fine <b>Value After Reset:</b> 0x9 <b>Exists:</b> Always
7:4	PclkDCAURMinFineDB	R/W	PclkDCAURMinFineDB: UR Min Fine <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
3:0	PclkDCAURMaxFineDB	R/W	PclkDCAURMaxFineDB: UR Max Fine <b>Value After Reset:</b> 0x6 <b>Exists:</b> Always

### 13.1.159 PclkDCANextFineOnCoarseDB

- **Name:** Sets the DCAFine value to use when Coarse changes
- **Description:** PclkDCANextFineOnCoarseDB: Sets the DCAFine value to use when Coarse changes. Used by PHYINIT.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0x807
- **Exists:** Always



**Table 13-163 Fields for Register: PclkDCANextFineOnCoarseDB**

Bits	Name	Memory Access	Description
15:12	PclkDCACoarseDecFineLLDB	R/W	PclkDCACoarseDecFineLLDB: Use this when coarse decrements in LL <b>Value After Reset:</b> 0xc <b>Exists:</b> Always
11:8	PclkDCACoarseIncFineLLDB	R/W	PclkDCACoarseIncFineLLDB: Use this when coarse increments in LL <b>Value After Reset:</b> 0xa <b>Exists:</b> Always
7:4	PclkDCACoarseDecFineURDB	R/W	PclkDCACoarseDecFineURDB: Use this when coarse decrements in UR <b>Value After Reset:</b> 0x2 <b>Exists:</b> Always
3:0	PclkDCACoarseIncFineURDB	R/W	PclkDCACoarseIncFineURDB: Use this when coarse increments in UR <b>Value After Reset:</b> 0x4 <b>Exists:</b> Always

### 13.1.160 PclkDCAFullSearchIVACDB

- **Name:** Initial DCA Fine value to use for full searches
- **Description:** PclkDCAFullSearchIVACDB: Initial DCA Fine value to use for full searches. Used by PHYINIT.
- **Size:** 8 bits
- **Offset:** (0x10000+(j<<12))+0x808
- **Exists:** Always



**Table 13-164 Fields for Register: PclkDCAFullSearchIVACDB**

Bits	Name	Memory Access	Description
7:4	PclkDCAFineIVMinDB	R/W	PclkDCAFineIVMinDB: Fine value for Samp 2 <b>Value After Reset:</b> 0xe <b>Exists:</b> Always
3:0	PclkDCAFineIVMaxDB	R/W	PclkDCAFineIVMaxDB: Fine value for Samp 1 <b>Value After Reset:</b> 0x6 <b>Exists:</b> Always

### 13.1.161 PclkDCASampDelayLCDLDB\_pX (for X = 0; X <= 3)

- **Description:** PclkDCASampDelayLCDLDB\_pX: Additional delay for SampVld generation in LCDL mode
- **Size:** 4 bits
- **Offset:** (0x10000+(j<<12))+0x80b+(X\*0x100000)
- **Exists:** Always

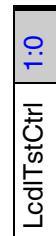


**Table 13-165 Fields for Register: PclkDCASampDelayLCDLDB\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
3:0	PclkDCASampDelayLCDLDB_p0	R/W	PclkDCASampDelayLCDLDB_p0: Set by PHYINIT. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.162 LcdITstCtrl

- **Description:** LcdITstCtrl: Reserved for PHY test firmware use.
- **Size:** 2 bits
- **Offset:** (0x10000+(j<<12))+0x884
- **Exists:** Always



**Table 13-166 Fields for Register: LcdITstCtrl**

Bits	Name	Memory Access	Description
1:0	LcdITstCtrl	R/W	LcdITstCtrl: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.163 DtsmErrCountNZ

- **Description:** DtsmErrCountNZ: Reserved for PHY training firmware use.
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0x9bf
- **Exists:** Always

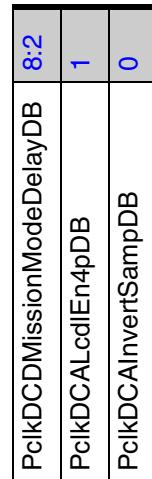


**Table 13-167 Fields for Register: DtsmErrCountNZ**

Bits	Name	Memory Access	Description
8:0	DtsmErrCountNZ	R	<p>DtsmErrCountNZ: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.1.164 PclkDCAStaticCtrl1DB\_pX (for X = 0; X <= 3)

- **Name:** Static bits for DCA control in DBYTE blocks
- **Description:** PclkDCAStaticCtrl1DB\_pX: Static bits for DCA control of DBYTE blocks. Used by PHYINIT.
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0xc03+(X\*0x100000)
- **Exists:** Always

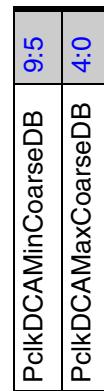


**Table 13-168 Fields for Register: PclkDCAStaticCtrl1DB\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
8:2	PclkDCDMissionModeDelayDB	R/W	<p>PclkDCDMissionModeDelayDB: Set by PHYINIT when PCLK DCA is using Comparator Indicates required delay between DCD_MODE-&gt;2 and first ClkSample Delay will [(2*VALUE)-1] DFICLKs. Reset VALUE is 4.</p> <p><b>Value After Reset:</b> 0x4</p> <p><b>Exists:</b> Always</p>
1	PclkDCALcdlEn4pDB	R/W	<p>PclkDCALcdlEn4pDB: Controls the EN4P pin of every DBYTE PclkDCA LcdlCalSeq</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
0	PclkDCAInvertSampDB	R/W	<p>PclkDCAInvertSampDB: 1 = Invert sample before using it. Should always be 0.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.165 PclkDCACoarseBoundDB

- **Name:** Sets limit values of DCA Coarse searches
- **Description:** PclkDCACoarseBoundDB: Coarse value Min/Max. Set by PHYINIT.
- **Size:** 10 bits
- **Offset:** (0x10000+(j<<12))+0xc1d
- **Exists:** Always



**Table 13-169 Fields for Register: PclkDCACoarseBoundDB**

Bits	Name	Memory Access	Description
9:5	PclkDCAMinCoarseDB	R/W	PclkDCAMinCoarseDB: Min Coarse value <b>Value After Reset:</b> 0x14 <b>Exists:</b> Always
4:0	PclkDCAMaxCoarseDB	R/W	PclkDCAMaxCoarseDB: Max Coarse value <b>Value After Reset:</b> 0x4 <b>Exists:</b> Always

### 13.1.166 PclkDCAMiscCtrlDB

- **Description:** PclkDCAMiscCtrlDB: Misc DCA control bits
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0xc1f
- **Exists:** Always



**Table 13-170 Fields for Register: PclkDCAMiscCtrlDB**

Bits	Name	Memory Access	Description
8:2	PclkDCAReservedDB	R/W	PclkDCAReservedDB: RFU <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
1	PclkDCDForceCkEnDB	R/W	PclkDCDForceCkEnDB: 1 = Drive the CkEn pin of DCD comparator high <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
0	PclkDCADitherModeDB	R/W	PclkDCADitherModeDB: Controls final DCA decision when dithering after Full Search 0 = Use Fine value after the Samp transition 1 = Use Fine value before the Samp transition <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.167 TrainingResultsSnap

- **Description:** TrainingResultsSnap: Enable for capture current contents of certain training registers.
- **Size:** 1 bit
- **Offset:** (0x10000+(j<<12))+0xd27
- **Exists:** Always



**Table 13-171 Fields for Register: TrainingResultsSnap**

Bits	Name	Memory Access	Description
0	TrainingResultsSnap	R/W	<p>TrainingResultsSnap: A 0 -&gt; 1 transition captures contents of csrTrainingCntr, csrDtsmErrCount, and csrDtsmGoodCount into the *_snap registers.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.168 PpgcChkCtrl

- **Description:** PpgcChkCtrl: Reserved for PHY training firmware use.
- **Size:** 1 bit
- **Offset:** (0x10000+(j<<12))+0xe00
- **Exists:** Always



**Table 13-172 Fields for Register: PpgcChkCtrl**

Bits	Name	Memory Access	Description
0	PpgcChkCtrl	R/W	PpgcChkCtrl: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.169 PpgcChkDbiCtrl

- **Description:** PpgcChkDbiCtrl: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe01
- **Exists:** Always



Table 13-173 Fields for Register: PpgcChkDbiCtrl

Bits	Name	Memory Access	Description
15:0	PpgcChkDbiCtrl	R/W	<p>PpgcChkDbiCtrl: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.170 PpgcChkDbiConfig

- **Description:** PpgcChkDbiConfig: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe02
- **Exists:** Always



**Table 13-174 Fields for Register: PpgcChkDbiConfig**

Bits	Name	Memory Access	Description
15:0	PpgcChkDbiConfig	R/W	PpgcChkDbiConfig: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.171 PpgcChkLaneMuxSel0

- **Description:** PpgcChkLaneMuxSel0: Reserved for PHY training firmware use.
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0xe03
- **Exists:** Always



**Table 13-175 Fields for Register: PpgcChkLaneMuxSel0**

Bits	Name	Memory Access	Description
8:0	PpgcChkLaneMuxSel0	R/W	<p>PpgcChkLaneMuxSel0: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.172 PpgcChkLaneMuxSel1

- **Description:** PpgcChkLaneMuxSel1: Reserved for PHY training firmware use.
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0xe04
- **Exists:** Always



**Table 13-176 Fields for Register: PpgcChkLaneMuxSel1**

Bits	Name	Memory Access	Description
8:0	PpgcChkLaneMuxSel1	R/W	<p>PpgcChkLaneMuxSel1: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.173 PpgcChkTxManipulationCtrl0

- **Description:** PpgcChkTxManipulationCtrl0: Reserved for PHY training firmware use.
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0xe08
- **Exists:** Always



**Table 13-177 Fields for Register: PpgcChkTxManipulationCtrl0**

Bits	Name	Memory Access	Description
8:0	PpgcChkTxManipulationCtrl0	R/W	<p>PpgcChkTxManipulationCtrl0: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.174 PpgcChkTxManipulationCtrl1

- **Description:** PpgcChkTxManipulationCtrl1: Reserved for PHY training firmware use.
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0xe09
- **Exists:** Always



**Table 13-178 Fields for Register: PpgcChkTxManipulationCtrl1**

Bits	Name	Memory Access	Description
8:0	PpgcChkTxManipulationCtrl1	R/W	<p>PpgcChkTxManipulationCtrl1: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.175 PpgcChkTxManipulationCtrl2

- **Description:** PpgcChkTxManipulationCtrl2: Reserved for PHY training firmware use.
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0xe0a
- **Exists:** Always



**Table 13-179 Fields for Register: PpgcChkTxManipulationCtrl2**

Bits	Name	Memory Access	Description
8:0	PpgcChkTxManipulationCtrl2	R/W	<p>PpgcChkTxManipulationCtrl2: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.176 PpgcChkTxManipulationPrbs9

- **Description:** PpgcChkTxManipulationPrbs9: Reserved for PHY training firmware use.
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0xe0b
- **Exists:** Always



**Table 13-180 Fields for Register: PpgcChkTxManipulationPrbs9**

Bits	Name	Memory Access	Description
8:0	PpgcChkTxManipulationPrbs9	R/W	<p>PpgcChkTxManipulationPrbs9: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.177 PpgcChkRxManipulationCtrl0

- **Description:** PpgcChkRxManipulationCtrl0: Reserved for PHY training firmware use.
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0xe0c
- **Exists:** Always



**Table 13-181 Fields for Register: PpgcChkRxManipulationCtrl0**

Bits	Name	Memory Access	Description
8:0	PpgcChkRxManipulationCtrl0	R/W	<p>PpgcChkRxManipulationCtrl0: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.178 PpgcChkRxManipulationCtrl1

- **Description:** PpgcChkRxManipulationCtrl1: Reserved for PHY training firmware use.
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0xe0d
- **Exists:** Always



**Table 13-182 Fields for Register: PpgcChkRxManipulationCtrl1**

Bits	Name	Memory Access	Description
8:0	PpgcChkRxManipulationCtrl1	R/W	<p>PpgcChkRxManipulationCtrl1: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.179 PpgcChkRxManipulationCtrl2

- **Description:** PpgcChkRxManipulationCtrl2: Reserved for PHY training firmware use.
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0xe0e
- **Exists:** Always



**Table 13-183 Fields for Register: PpgcChkRxManipulationCtrl2**

Bits	Name	Memory Access	Description
8:0	PpgcChkRxManipulationCtrl2	R/W	<p>PpgcChkRxManipulationCtrl2: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.180 PpgcChkRxManipluationPrbs9

- **Description:** PpgcChkRxManipluationPrbs9: Reserved for PHY training firmware use.
- **Size:** 9 bits
- **Offset:** (0x10000+(j<<12))+0xe0f
- **Exists:** Always



**Table 13-184 Fields for Register: PpgcChkRxManipluationPrbs9**

Bits	Name	Memory Access	Description
8:0	PpgcChkRxManipluationPrbs9	R/W	<p>PpgcChkRxManipluationPrbs9: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.181 PclkDCAClkGaterEnDB

- **Description:** PclkDCAClkGaterEnDB: 1 = Clk to DCA FSMs is enabled
- **Size:** 1 bit
- **Offset:** (0x10000+(j<<12))+0xe1f
- **Exists:** Always



**Table 13-185 Fields for Register: PclkDCAClkGaterEnDB**

Bits	Name	Memory Access	Description
0	PclkDCAClkGaterEnDB	R/W	PclkDCAClkGaterEnDB: 1 = Clk to DCA FSMs is enabled <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always

### 13.1.182 Prbs0ChkModeSel

- **Description:** Prbs0ChkModeSel: Reserved for PHY training firmware use.
- **Size:** 2 bits
- **Offset:** (0x10000+(j<<12))+0xe20
- **Exists:** Always



**Table 13-186 Fields for Register: Prbs0ChkModeSel**

Bits	Name	Memory Access	Description
1:0	Prbs0ChkModeSel	R/W	Prbs0ChkModeSel: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.183 Prbs0ChkUiMuxSel

- Description:** Prbs0ChkUiMuxSel: Reserved for PHY training firmware use.
- Size:** 2 bits
- Offset:** (0x10000+(j<<12))+0xe21
- Exists:** Always



**Table 13-187 Fields for Register: Prbs0ChkUiMuxSel**

Bits	Name	Memory Access	Description
1:0	Prbs0ChkUiMuxSel	R/W	<p>Prbs0ChkUiMuxSel: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.184 Prbs0ChkTapDly0

- **Description:** Prbs0ChkTapDly0: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe22
- **Exists:** Always



**Table 13-188 Fields for Register: Prbs0ChkTapDly0**

Bits	Name	Memory Access	Description
15:0	Prbs0ChkTapDly0	R/W	Prbs0ChkTapDly0: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.185 Prbs0ChkTapDly1

- Description:** Prbs0ChkTapDly1: Reserved for PHY training firmware use.
- Size:** 16 bits
- Offset:** (0x10000+(j<<12))+0xe23
- Exists:** Always



Table 13-189 Fields for Register: Prbs0ChkTapDly1

Bits	Name	Memory Access	Description
15:0	Prbs0ChkTapDly1	R/W	<p>Prbs0ChkTapDly1: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.186 Prbs0ChkTapDly2

- **Description:** Prbs0ChkTapDly2: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe24
- **Exists:** Always



**Table 13-190 Fields for Register: Prbs0ChkTapDly2**

Bits	Name	Memory Access	Description
15:0	Prbs0ChkTapDly2	R/W	Prbs0ChkTapDly2: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.187 Prbs0ChkTapDly3

- **Description:** Prbs0ChkTapDly3: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe25
- **Exists:** Always



**Table 13-191 Fields for Register: Prbs0ChkTapDly3**

Bits	Name	Memory Access	Description
15:0	Prbs0ChkTapDly3	R/W	<p>Prbs0ChkTapDly3: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.188 Prbs0ChkTapDly4

- **Description:** Prbs0ChkTapDly4: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe26
- **Exists:** Always



**Table 13-192 Fields for Register: Prbs0ChkTapDly4**

Bits	Name	Memory Access	Description
15:0	Prbs0ChkTapDly4	R/W	Prbs0ChkTapDly4: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.189 Prbs0ChkTapDly5

- Description:** Prbs0ChkTapDly5: Reserved for PHY training firmware use.
- Size:** 16 bits
- Offset:** (0x10000+(j<<12))+0xe27
- Exists:** Always



Table 13-193 Fields for Register: Prbs0ChkTapDly5

Bits	Name	Memory Access	Description
15:0	Prbs0ChkTapDly5	R/W	<p>Prbs0ChkTapDly5: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.190 Prbs0ChkTapDly6

- **Description:** Prbs0ChkTapDly6: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe28
- **Exists:** Always



**Table 13-194 Fields for Register: Prbs0ChkTapDly6**

Bits	Name	Memory Access	Description
15:0	Prbs0ChkTapDly6	R/W	Prbs0ChkTapDly6: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.191 Prbs0ChkTapDly7

- Description:** Prbs0ChkTapDly7: Reserved for PHY training firmware use.
- Size:** 16 bits
- Offset:** (0x10000+(j<<12))+0xe29
- Exists:** Always



**Table 13-195 Fields for Register: Prbs0ChkTapDly7**

Bits	Name	Memory Access	Description
15:0	Prbs0ChkTapDly7	R/W	<p>Prbs0ChkTapDly7: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.192 Prbs0ChkStateLo

- **Description:** Prbs0ChkStateLo: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:**  $(0x10000 + (j << 12)) + 0xe2a$
- **Exists:** Always



**Table 13-196 Fields for Register: Prbs0ChkStateLo**

Bits	Name	Memory Access	Description
15:0	Prbs0ChkStateLo	R/W	Prbs0ChkStateLo: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.193 Prbs0ChkStateHi

- **Description:** Prbs0ChkStateHi: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe2b
- **Exists:** Always



Table 13-197 Fields for Register: Prbs0ChkStateHi

Bits	Name	Memory Access	Description
15:0	Prbs0ChkStateHi	R/W	<p>Prbs0ChkStateHi: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.194 Prbs1ChkModeSel

- **Description:** Prbs1ChkModeSel: Reserved for PHY training firmware use.
- **Size:** 2 bits
- **Offset:** (0x10000+(j<<12))+0xe30
- **Exists:** Always



**Table 13-198 Fields for Register: Prbs1ChkModeSel**

Bits	Name	Memory Access	Description
1:0	Prbs1ChkModeSel	R/W	Prbs1ChkModeSel: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.195 Prbs1ChkUiMuxSel

- **Description:** Prbs1ChkUiMuxSel: Reserved for PHY training firmware use.
- **Size:** 2 bits
- **Offset:** (0x10000+(j<<12))+0xe31
- **Exists:** Always



**Table 13-199 Fields for Register: Prbs1ChkUiMuxSel**

Bits	Name	Memory Access	Description
1:0	Prbs1ChkUiMuxSel	R/W	<p>Prbs1ChkUiMuxSel: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.196 Prbs1ChkTapDly0

- **Description:** Prbs1ChkTapDly0: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe32
- **Exists:** Always



**Table 13-200 Fields for Register: Prbs1ChkTapDly0**

Bits	Name	Memory Access	Description
15:0	Prbs1ChkTapDly0	R/W	Prbs1ChkTapDly0: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.197 Prbs1ChkTapDly1

- Description:** Prbs1ChkTapDly1: Reserved for PHY training firmware use.
- Size:** 16 bits
- Offset:**  $(0x10000 + (j << 12)) + 0xe33$
- Exists:** Always



Table 13-201 Fields for Register: Prbs1ChkTapDly1

Bits	Name	Memory Access	Description
15:0	Prbs1ChkTapDly1	R/W	<p>Prbs1ChkTapDly1: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.198 Prbs1ChkTapDly2

- **Description:** Prbs1ChkTapDly2: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe34
- **Exists:** Always



**Table 13-202 Fields for Register: Prbs1ChkTapDly2**

Bits	Name	Memory Access	Description
15:0	Prbs1ChkTapDly2	R/W	Prbs1ChkTapDly2: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.199 Prbs1ChkTapDly3

- **Description:** Prbs1ChkTapDly3: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe35
- **Exists:** Always



**Table 13-203 Fields for Register: Prbs1ChkTapDly3**

Bits	Name	Memory Access	Description
15:0	Prbs1ChkTapDly3	R/W	<p>Prbs1ChkTapDly3: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.200 Prbs1ChkTapDly4

- **Description:** Prbs1ChkTapDly4: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe36
- **Exists:** Always



**Table 13-204 Fields for Register: Prbs1ChkTapDly4**

Bits	Name	Memory Access	Description
15:0	Prbs1ChkTapDly4	R/W	Prbs1ChkTapDly4: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.201 Prbs1ChkTapDly5

- Description:** Prbs1ChkTapDly5: Reserved for PHY training firmware use.
- Size:** 16 bits
- Offset:** (0x10000+(j<<12))+0xe37
- Exists:** Always



**Table 13-205 Fields for Register: Prbs1ChkTapDly5**

Bits	Name	Memory Access	Description
15:0	Prbs1ChkTapDly5	R/W	<p>Prbs1ChkTapDly5: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.202 Prbs1ChkTapDly6

- **Description:** Prbs1ChkTapDly6: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe38
- **Exists:** Always



**Table 13-206 Fields for Register: Prbs1ChkTapDly6**

Bits	Name	Memory Access	Description
15:0	Prbs1ChkTapDly6	R/W	Prbs1ChkTapDly6: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.203 Prbs1ChkTapDly7

- Description:** Prbs1ChkTapDly7: Reserved for PHY training firmware use.
- Size:** 16 bits
- Offset:**  $(0x10000 + (j << 12)) + 0xe39$
- Exists:** Always



**Table 13-207 Fields for Register: Prbs1ChkTapDly7**

Bits	Name	Memory Access	Description
15:0	Prbs1ChkTapDly7	R/W	<p>Prbs1ChkTapDly7: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.204 Prbs1ChkStateLo

- **Description:** Prbs1ChkStateLo: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:**  $(0x10000 + (j \ll 12)) + 0xe3a$
- **Exists:** Always



**Table 13-208 Fields for Register: Prbs1ChkStateLo**

Bits	Name	Memory Access	Description
15:0	Prbs1ChkStateLo	R/W	Prbs1ChkStateLo: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.205 Prbs1ChkStateHi

- **Description:** Prbs1ChkStateHi: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe3b
- **Exists:** Always



Table 13-209 Fields for Register: Prbs1ChkStateHi

Bits	Name	Memory Access	Description
15:0	Prbs1ChkStateHi	R/W	Prbs1ChkStateHi: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.206 Prbs2ChkModeSel

- **Description:** Prbs2ChkModeSel: Reserved for PHY training firmware use.
- **Size:** 2 bits
- **Offset:** (0x10000+(j<<12))+0xe40
- **Exists:** Always



**Table 13-210 Fields for Register: Prbs2ChkModeSel**

Bits	Name	Memory Access	Description
1:0	Prbs2ChkModeSel	R/W	Prbs2ChkModeSel: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.207 Prbs2ChkUiMuxSel

- **Description:** Prbs2ChkUiMuxSel: Reserved for PHY training firmware use.
- **Size:** 2 bits
- **Offset:** (0x10000+(j<<12))+0xe41
- **Exists:** Always



**Table 13-211 Fields for Register: Prbs2ChkUiMuxSel**

Bits	Name	Memory Access	Description
1:0	Prbs2ChkUiMuxSel	R/W	Prbs2ChkUiMuxSel: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.208 Prbs2ChkTapDly0

- **Description:** Prbs2ChkTapDly0: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe42
- **Exists:** Always



**Table 13-212 Fields for Register: Prbs2ChkTapDly0**

Bits	Name	Memory Access	Description
15:0	Prbs2ChkTapDly0	R/W	Prbs2ChkTapDly0: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.209 Prbs2ChkTapDly1

- Description:** Prbs2ChkTapDly1: Reserved for PHY training firmware use.
- Size:** 16 bits
- Offset:**  $(0x10000 + (j << 12)) + 0xe43$
- Exists:** Always



**Table 13-213 Fields for Register: Prbs2ChkTapDly1**

Bits	Name	Memory Access	Description
15:0	Prbs2ChkTapDly1	R/W	<p>Prbs2ChkTapDly1: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.210 Prbs2ChkTapDly2

- **Description:** Prbs2ChkTapDly2: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe44
- **Exists:** Always



**Table 13-214 Fields for Register: Prbs2ChkTapDly2**

Bits	Name	Memory Access	Description
15:0	Prbs2ChkTapDly2	R/W	Prbs2ChkTapDly2: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.211 Prbs2ChkTapDly3

- **Description:** Prbs2ChkTapDly3: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe45
- **Exists:** Always



**Table 13-215 Fields for Register: Prbs2ChkTapDly3**

Bits	Name	Memory Access	Description
15:0	Prbs2ChkTapDly3	R/W	<p>Prbs2ChkTapDly3: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.212 Prbs2ChkTapDly4

- **Description:** Prbs2ChkTapDly4: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe46
- **Exists:** Always



**Table 13-216 Fields for Register: Prbs2ChkTapDly4**

Bits	Name	Memory Access	Description
15:0	Prbs2ChkTapDly4	R/W	Prbs2ChkTapDly4: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.213 Prbs2ChkTapDly5

- Description:** Prbs2ChkTapDly5: Reserved for PHY training firmware use.
- Size:** 16 bits
- Offset:** (0x10000+(j<<12))+0xe47
- Exists:** Always



Table 13-217 Fields for Register: Prbs2ChkTapDly5

Bits	Name	Memory Access	Description
15:0	Prbs2ChkTapDly5	R/W	<p>Prbs2ChkTapDly5: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.214 Prbs2ChkTapDly6

- **Description:** Prbs2ChkTapDly6: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe48
- **Exists:** Always



**Table 13-218 Fields for Register: Prbs2ChkTapDly6**

Bits	Name	Memory Access	Description
15:0	Prbs2ChkTapDly6	R/W	Prbs2ChkTapDly6: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.215 Prbs2ChkTapDly7

- Description:** Prbs2ChkTapDly7: Reserved for PHY training firmware use.
- Size:** 16 bits
- Offset:**  $(0x10000 + (j << 12)) + 0xe49$
- Exists:** Always



**Table 13-219 Fields for Register: Prbs2ChkTapDly7**

Bits	Name	Memory Access	Description
15:0	Prbs2ChkTapDly7	R/W	<p>Prbs2ChkTapDly7: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.216 Prbs2ChkStateLo

- **Description:** Prbs2ChkStateLo: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe4a
- **Exists:** Always



**Table 13-220 Fields for Register: Prbs2ChkStateLo**

Bits	Name	Memory Access	Description
15:0	Prbs2ChkStateLo	R/W	Prbs2ChkStateLo: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.217 Prbs2ChkStateHi

- **Description:** Prbs2ChkStateHi: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xe4b
- **Exists:** Always



Table 13-221 Fields for Register: Prbs2ChkStateHi

Bits	Name	Memory Access	Description
15:0	Prbs2ChkStateHi	R/W	<p>Prbs2ChkStateHi: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.1.218 PpgcChkFltCfg0

- **Description:** PpgcChkFltCfg0: Reserved for PHY training firmware use.
- **Size:** 5 bits
- **Offset:** (0x10000+(j<<12))+0xea0
- **Exists:** Always



**Table 13-222 Fields for Register: PpgcChkFltCfg0**

Bits	Name	Memory Access	Description
4:0	PpgcChkFltCfg0	R/W	PpgcChkFltCfg0: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.219 PpgcChkFltCfg1

- **Description:** PpgcChkFltCfg1: Reserved for PHY training firmware use.
- **Size:** 5 bits
- **Offset:** (0x10000+(j<<12))+0xea1
- **Exists:** Always



**Table 13-223 Fields for Register: PpgcChkFltCfg1**

Bits	Name	Memory Access	Description
4:0	PpgcChkFltCfg1	R/W	PpgcChkFltCfg1: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.220 PpgcChkFltCfg2

- **Description:** PpgcChkFltCfg2: Reserved for PHY training firmware use.
- **Size:** 4 bits
- **Offset:** (0x10000+(j<<12))+0xea2
- **Exists:** Always



**Table 13-224 Fields for Register: PpgcChkFltCfg2**

Bits	Name	Memory Access	Description
3:0	PpgcChkFltCfg2	R/W	PpgcChkFltCfg2: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.221 PpgcChkMskPat0

- **Description:** PpgcChkMskPat0: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xea4
- **Exists:** Always



Table 13-225 Fields for Register: PpgcChkMskPat0

Bits	Name	Memory Access	Description
15:0	PpgcChkMskPat0	R/W	PpgcChkMskPat0: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.222 PpgcChkMskPat1

- **Description:** PpgcChkMskPat1: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xea5
- **Exists:** Always



**Table 13-226 Fields for Register: PpgcChkMskPat1**

Bits	Name	Memory Access	Description
15:0	PpgcChkMskPat1	R/W	PpgcChkMskPat1: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.223 PpgcChkMskPat2

- **Description:** PpgcChkMskPat2: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xea6
- **Exists:** Always



**Table 13-227 Fields for Register: PpgcChkMskPat2**

Bits	Name	Memory Access	Description
15:0	PpgcChkMskPat2	R/W	PpgcChkMskPat2: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.1.224 PpgcChkMskPat3

- **Description:** PpgcChkMskPat3: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** (0x10000+(j<<12))+0xea7
- **Exists:** Always



**Table 13-228 Fields for Register: PpgcChkMskPat3**

Bits	Name	Memory Access	Description
15:0	PpgcChkMskPat3	R/W	PpgcChkMskPat3: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

## 13.2 **DWC\_DDRPHYA\_MASTERj\_Pk Registers**

### 13.2.1 DfiFreqRatio\_pX (for X = 0; X <= 3)

- **Description:** DfiFreqRatio\_pX: DFI Frequency Ratio
- **Size:** 2 bits
- **Offset:** 0x20000+0x0+(X\*0x100000)
- **Exists:** Always



**Table 13-229 Fields for Register: DfiFreqRatio\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
1:0	DfiFreqRatio_p0	R/W	<p>DfiFreqRatio_p0: DFI Frequency Ratio</p> <ul style="list-style-type: none"> <li>■ 2'b00, 2'b11 - Reserved</li> <li>■ 2'b01 - 1:2 Ratio</li> <li>■ 2'b10 - 1:4 Ratio</li> <li>■ In LPDDR4X, This is the Ratio between DfiClk : Memory Clock</li> <li>■ In LPDDR5, This is the Ratio between DfiClk : WCK Clock</li> </ul> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

### 13.2.2 PubDxPowerDownControl

- **Description:** PubDxPowerDownControl: Bits to turn off the PUB\_DX
- **Size:** 2 bits
- **Offset:** 0x20000+0x1
- **Exists:** Always

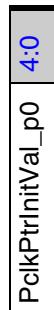


**Table 13-230 Fields for Register: PubDxPowerDownControl**

Bits	Name	Memory Access	Description
1	D5DxPowerDown	R/W	D5DxPowerDown: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
0	LP5DxPowerDown	R/W	LP5DxPowerDown: When set, powers down the LP5-specific PUB modules <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.2.3 PclkPtrInitVal\_pX (for X = 0; X <= 3)

- **Description:** PclkPtrInitVal\_pX: PAC Hard Macro Command FIFO ReadPointer Initial Value
- **Size:** 5 bits
- **Offset:** 0x20000+0x2+(X\*0x100000)
- **Exists:** Always



**Table 13-231 Fields for Register: PclkPtrInitVal\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
4:0	PclkPtrInitVal_p0	R/W	<p>PclkPtrInitVal_p0: Controls the phase offset between read and write pointers of the PAC hard macro command FIFO. The units of this register is 0.5 x PLL pllout_clk clock period. The pointer separation should be chosen to compensate for all sources of skew and drift of the PHY DFICLK and PCLK networks. This CSR must be programmed in Step C of the PHY Initialization sequence. Both csrPclkPtrInitVal and csrHMPclkPtrInitVal must always be programmed to same matching value.</p> <p><b>Value After Reset:</b> 0x2</p> <p><b>Exists:</b> Always</p>

### 13.2.4 CmdFifoWrModeMaster\_pX (for X = 0; X <= 3)

- **Description:** CmdFifoWrModeMaster\_pX: PAC Hard Macro Command FIFO Write Mode Value
- **Size:** 1 bit
- **Offset:** 0x20000+0x3+(X\*0x100000)
- **Exists:** Always



**Table 13-232 Fields for Register: CmdFifoWrModeMaster\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
0	CmdFifoWrModeMaster_p0	R/W	<p>CmdFifoWrModeMaster_p0: Selects the valid width for the PAC hard macro Command FIFO inputs (inputs are always 8 bit wide). Valid values are:</p> <ul style="list-style-type: none"> <li>■ 1'b0: Only lower 4 bits will be written in Fifo on each DfiClk.</li> <li>■ 1'b1: All 8 bits will be written on each DfiClk.</li> </ul> <p>All other values are not supported. This CSR must be programmed in Step C of the PHY Initialization sequence</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.2.5 MTestDtoCtrl

- **Name:** Enables MTestCombo output on core-side copy
- **Description:** MTestDtoCtrl: If set, allows the MTestCombo observability signal to be routed out core-side through the \_dto pin
- **Size:** 1 bit
- **Offset:** 0x20000+0x4
- **Exists:** Always

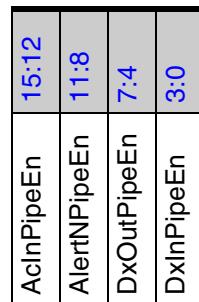


**Table 13-233 Fields for Register: MTestDtoCtrl**

Bits	Name	Memory Access	Description
0	MTESTdtoEn	R/W	<p>MTESTdtoEn: This field is programmed as follows:</p> <ul style="list-style-type: none"> <li>■ MTESTdtoEn==[0], _dto will be squelched (0)</li> <li>■ MTESTdtoEn==[1], _dto will reflect the observability signal multiplexed on MTestCombo</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.2.6 PipeCtl\_pX (for X = 0; X <= 3)

- **Name:** Delay Enables for PIPE block
- **Description:** PipeCtl\_pX: When set, each bit enables a stage of delay for the signals passing through the PIPE block
- **Size:** 16 bits
- **Offset:** 0x20000+0x5+(X\*0x100000)
- **Exists:** Always



**Table 13-234 Fields for Register: PipeCtl\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:12	AcInPipeEn	R/W	<p>AcInPipeEn: Controls the PIPE stages for these signals  <code>dfi_reset_n</code>, <code>dfi*_address</code>, <code>dfi*_ctrlupd_req</code>, <code>dfi*_ctrlupd_type</code>,  <code>dfi*_phyupd_ack</code>, <code>dfi*_dram_clk_disable</code>, <code>dfi*_freq_fsp</code>,  <code>dfi*_freq_ratio</code>, <code>dfi*_frequency</code>, <code>dfi*_init_start</code>,  <code>dfi*_phymstr_ack</code>, <code>dfi*_cs</code>, <code>dfi*_lp_ctrl_req</code>,  <code>dfi*_lp_ctrl_wakeup</code>, <code>dfi*_lp_data_req</code>, <code>dfi*_lp_data_wakeup</code></p> <p>Only bit 12 is used</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
11:8	AlertNPipeEn	R/W	<p>AlertNPipeEn: Reserved</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
7:4	DxOutPipeEn	R/W	<p>DxOutPipeEn: Controls the PIPE stages for <code>dfi*_rddata_*</code> signals not controlled in DxInPipeEn Only bit 4 is used</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
3:0	DxInPipeEn	R/W	<p>DxInPipeEn: Controls the PIPE stages for these signals  <code>dfi*_wrdata</code>, <code>dfi*_wrdata_cs</code>, <code>dfi*_wrdata_en</code>,  <code>dfi*_wrdata_mask</code>, <code>dfi*_wck_cs</code>, <code>dfi*_wck_en</code>, <code>dfi*_rddata_en</code></p> <p>Only bit 0 is used</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.2.7 LpDqPhaseDisable

- **Description:** LpDqPhaseDisable: Disables the PAC hard macro clock when in LP state with clocks stopped
- **Size:** 5 bits
- **Offset:** 0x20000+0x6
- **Exists:** Always



**Table 13-235 Fields for Register: LpDqPhaseDisable**

Bits	Name	Memory Access	Description
4:0	LpDqPhaseDisable	R/W	<p>LpDqPhaseDisable: Disables the PAC hard macro clock to designated sections when in LP Ctrl+Data state with clocks stopped Bit 0 Disables clock to HMDBYTEs Bit 1 Disables clock to PUB_DX Bit 2 Disables clock to HMACs Bit 3 Disables clock to PUB_AC Bit 4 Disables clock to PIPE if both channels are in dfi_lp mode</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.2.8 PubDbyteDisable

- **Description:** PubDbyteDisable: Forces PUB logic for a dbyte into low-power/disabled mode
- **Size:** 4 bits
- **Offset:** 0x20000+0x7
- **Exists:** Always



**Table 13-236 Fields for Register: PubDbyteDisable**

Bits	Name	Memory Access	Description
3:0	PubDbyteDisable	R/W	<p>PubDbyteDisable: Each bit corresponds to the Dbyte with the matching index. When a bit is set, the corresponding PUB DBYTE logic is moved to the same low-power state as if a dfi_lp event had occurred. This should be set only after setting csrDByteDisable, and cleared only before clearing csrDByteDisable.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.2.9 CPclkDivRatio\_pX (for X = 0; X <= 3)

- **Name:** Pclk clock divider ratios
- **Description:** CPclkDivRatio\_pX: Individual fields to specify the Pclk\_Ca and Pclk\_Dq divide ratios, in the PAC hard macro clock dividers. The PLL output clock is divided by the divider value specified in these fields and forwarded to the IO slices as Pclk\_Ca0, Pclk\_Ca1, Pclk\_Dq0, Pclk\_Dq1. This CSR fields must be programmed in Step C of the PHY Initialization sequence
- **Size:** 14 bits
- **Offset:** 0x20000+0xb+(X\*0x100000)
- **Exists:** Always

CPclkDivDq1	13:12
ReservedCPclkDivDq0	11:10
CPclkDivDq0	9:8
ReservedCPclkDivCa1	7:6
CPclkDivCa1	5:4
ReservedCPclkDivCa0	3:2
CPclkDivCa0	1:0

Table 13-237 Fields for Register: CPclkDivRatio\_pX (for X = 0; X <= 3)

Bits	Name	Memory Access	Description
13:12	CPclkDivDq1	R/W	<p>CPclkDivDq1: Pattern to specify the divide ratio for Pclk_Dq1, going to all CA slices for channel 1</p> <ul style="list-style-type: none"> <li>■ 2'b00 - 0, Pclk_Dq1 forced 0</li> <li>■ 2'b01 - Divide by 1, Pclk_Dq1 = PLL output clock / 1</li> <li>■ 2'b10 - Divide by 2, Pclk_Dq1 = PLL output clock / 2</li> <li>■ 2'b11 - Illegal</li> </ul> <p><b>Value After Reset:</b> 0x1  <b>Exists:</b> Always</p>
11:10	ReservedCPclkDivDq0	R/W	<p>ReservedCPclkDivDq0: Reserved for future.</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
9:8	CPclkDivDq0	R/W	<p>CPclkDivDq0: Pattern to specify the divide ratio for Pclk_Dq0, going to all CA slices for channel 0</p> <ul style="list-style-type: none"> <li>■ 2'b00 - 0, Pclk_Dq0 forced 0</li> <li>■ 2'b01 - Divide by 1, Pclk_Dq0 = PLL output clock / 1</li> <li>■ 2'b10 - Divide by 2, Pclk_Dq0 = PLL output clock / 2</li> <li>■ 2'b11 - Illegal</li> </ul> <p><b>Value After Reset:</b> 0x1  <b>Exists:</b> Always</p>
7:6	ReservedCPclkDivCa1	R/W	<p>ReservedCPclkDivCa1: Reserved for future.</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
5:4	CPclkDivCa1	R/W	<p>CPclkDivCa1: Pattern to specify the divide ratio for Pclk_Ca1, going to all CA slices for channel 1</p> <ul style="list-style-type: none"> <li>■ 2'b00 - 0, Pclk_Ca1 forced 0</li> <li>■ 2'b01 - Divide by 1, Pclk_Ca1 = PLL output clock / 1</li> <li>■ 2'b10 - Divide by 2, Pclk_Ca1 = PLL output clock / 2</li> <li>■ 2'b11 - Illegal</li> </ul> <p><b>Value After Reset:</b> 0x1  <b>Exists:</b> Always</p>
3:2	ReservedCPclkDivCa0	R/W	<p>ReservedCPclkDivCa0: Reserved for future.</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
1:0	CPclkDivCa0	R/W	<p>CPclkDivCa0: Pattern to specify the divide ratio for Pclk_Ca0, going to all CA slices for channel 0</p> <ul style="list-style-type: none"> <li>■ 2'b00 - 0, Pclk_Ca0 forced 0</li> <li>■ 2'b01 - Divide by 1, Pclk_Ca0 = PLL output clock / 1</li> <li>■ 2'b10 - Divide by 2, Pclk_Ca0 = PLL output clock / 2</li> <li>■ 2'b11 - Illegal</li> </ul> <p><b>Value After Reset:</b> 0x1  <b>Exists:</b> Always</p>

### 13.2.10 PipeNetDis

- **Description:** PipeNetDis: Power Saving feature
- **Size:** 1 bit
- **Offset:** 0x20000+0xc
- **Exists:** Always



**Table 13-238 Fields for Register: PipeNetDis**

Bits	Name	Memory Access	Description
0	PipeNetDis	R/W	<p>PipeNetDis: For designs where All *PipeEn are always 0, set this to 1 to turn off the clock network to the mux flops in the PIPE block</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.2.11 MiscPipeEn

- **Description:** MiscPipeEn: Deprecated
- **Size:** 2 bits
- **Offset:** 0x20000+0xf
- **Exists:** Always



Table 13-239 Fields for Register: MiscPipeEn

Bits	Name	Memory Access	Description
1:0	MiscPipeEn	R/W	<p>MiscPipeEn: Deprecated</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.2.12 MtestMuxSelCMOS

- **Description:** MtestMuxSelCMOS: Digital Observation Pin control for HMCMOS
- **Size:** 5 bits
- **Offset:** 0x20000+0x10
- **Exists:** Always



**Table 13-240 Fields for Register: MtestMuxSelCMOS**

Bits	Name	Memory Access	Description
4:0	MtestMuxSelCMOS	R/W	<p>MtestMuxSelCMOS: Controls for the mux for asynchronous data to the Digital Observation Pin for CMOSX2_Top Encoding 5'h0 causes this chiplet to drive 0, (allowing flat 'OR' of pass-through information. A non-zero encoding selects one bit from the 32 bit MtestMux. Detailed tables are in the PUB Databook.<b>Note:</b>See PUB documentation for how, or if, the Digital Observation Pin is mapped to a physical bump in this configuration.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.2.13 EnRxDqsTracking\_pX (for X = 0; X <= 3)

- **Name:** Mode controls for fine RxEn timing adjustment during mission-mode operation.
- **Description:** EnRxDqsTracking\_pX: Enable tracking of the incoming read DQS to modify the RxEn timing.
- **Size:** 4 bits
- **Offset:** 0x20000+0x19+(X\*0x100000)
- **Exists:** Always



Table 13-241 Fields for Register: EnRxDqsTracking\_pX (for X = 0; X <= 3)

Bits	Name	Memory Access	Description
3	EnDqsSampNegRxEn	R/W	<p>EnDqsSampNegRxEn: Enable the sampling of read DQS by the deasserting edge of RxEn by a phase-detector in the dbyte read-state-machines This control may be used during training while RxEnTrain=1 or during mission-mode while csr EnRxDqsTracking=1</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
2	DqsSampNegRxEnSense	R/W	<p>DqsSampNegRxEnSense: Assert to invert the phase-detector value as seen by the tracking logic. If csr RxEnDlyTg is trained to be 2UI before the posedge of DQS_T, then set to 1. If csr RxEnDlyTg is trained to be 1UI before the posedge of DQS_T, then set to 0.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
1	EnDqsSampNegRxEnPPT	R/W	<p>EnDqsSampNegRxEnPPT: Used to enable the DBYTE phase-detector that uses the negedge DQS_t and the negedge of RxEn_async. This phase-detector is enabled if EnDqsSampNegRxEnPPT=1 or if EnDqsSampNegRxEn=1. May only be written while the DBYTE is idle. This phase-detector might be enabled to run all the time, using EnDqsSampNegRxEn, if there is analog reason to do so, with the understanding that there is a small power cost.</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
0	EnRxDqsTrackingPipe	R/W	<p>EnRxDqsTrackingPipe: Enables the PUB pipeline to change RxEn timing. Requires the DBYTE RSM negedge RxEn_async phase-detector to be enabled.</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.2.14 MtestMuxSel

- **Description:** MtestMuxSel: Digital Observation Pin control
- **Size:** 10 bits
- **Offset:** 0x20000+0x1a
- **Exists:** Always

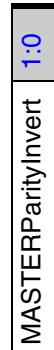


**Table 13-242 Fields for Register: MtestMuxSel**

Bits	Name	Memory Access	Description
9:0	MtestMuxSel	R/W	<p>MtestMuxSel: Controls for the mux for asynchronous data to the Digital Observation Pin.</p> <ul style="list-style-type: none"> <li>■ Encoding 9'h0 causes this chiplet to drive 0, (allowing flat 'OR' of pass-through information).</li> <li>■ MtestMuxSel[4:0] - Lower 5 bits selects one bit from the 32 bit MtestMux in each section or slice.</li> </ul> <p>Detailed tables are in the PUB Databook for PUB sections (AC, MASTER, etc.)</p> <ul style="list-style-type: none"> <li>■ MtestMuxSel[8:5] - Where more than one MtestMux exists, non-zero values select the outputs of the additional Mtest-Muxes,           <ul style="list-style-type: none"> <li>□ DBYTE MtestMuxSel[6:5]=2'h0 for Mux-A and 2'h1 for Mux-B and 2'h2 for Mux-C.</li> <li>□ For all other slaves, MtestMuxSel[8:5] are unused</li> </ul> </li> </ul> <p><b>Note:</b> See the PUB Databook for how, or if, the Digital Observation Pin is mapped to a physical bump in this configuration.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.2.15 MASTERParityInvert

- **Description:** MASTERParityInvert: Invert APB Parity for register slave PUB\_PAC
- **Size:** 2 bits
- **Offset:** 0x20000+0x4d
- **Exists:** Always



**Table 13-243 Fields for Register: MASTERParityInvert**

Bits	Name	Memory Access	Description
1:0	MASTERParityInvert	R/W	<p>MASTERParityInvert: Invert APB Parity for register slave PUB_PAC. As required for Automotive. NOTE: This register should be used only for test. Set the bits for only one slave at a time. When bits are set for a particular slave. APB Reads of only that slave are valid. Bit 0 applies to [7:0] Bit 1 applies to [15:8] In HMZCAL, When ZcalStopClk=1, this csr can not be accessed.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.2.16 DfiMode

- **Name:** Enables for update and low-power interfaces for DFI0 and DFI1
- **Description:** DfiMode: Each bit enables the Phy Update, Phy Master Update, and low power logic for the respective interfaces
- **Size:** 2 bits
- **Offset:** 0x20000+0x51
- **Exists:** Always



**Table 13-244 Fields for Register: DfiMode**

Bits	Name	Memory Access	Description
1	Dfi1Enable	R/W	<p>Dfi1Enable: Enables operation for the PHY logic associated with DFI1</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>
0	Dfi0Enable	R/W	<p>Dfi0Enable: Enables operation for the PHY logic associated with DFI0</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

### 13.2.17 MtestPgmlInfo

- **Description:** MtestPgmlInfo: Digital Observation Pin program info for debug
- **Size:** 1 bit
- **Offset:** 0x20000+0x52
- **Exists:** Always



**Table 13-245 Fields for Register: MtestPgmlInfo**

Bits	Name	Memory Access	Description
0	MtestPgmlInfo	R/W	<p>MtestPgmlInfo: The value of this csr may be driven onto the Digital Observation Pin. It has no other hardware effect.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.2.18 PhyTID

- **Description:** PhyTID: PHY Technology ID Register
- **Size:** 16 bits
- **Offset:** 0x20000+0x55
- **Exists:** Always



Table 13-246 Fields for Register: PhyTID

Bits	Name	Memory Access	Description
15:0	PhyTID	R/W	<p>PhyTID: This register is a placeholder to store technology-specific information</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.2.19 DbyteRxEnTrain

- **Description:** DbyteRxEnTrain: Reserved for PHY training firmware use.
- **Size:** 2 bits
- **Offset:** 0x20000+0x59
- **Exists:** Always



**Table 13-247 Fields for Register: DbyteRxEnTrain**

Bits	Name	Memory Access	Description
1:0	DbyteRxEnTrain	R/W	DbyteRxEnTrain: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.2.20 PUBMODE

- **Name:** PUBMODE - HWT Mux Select
- **Description:** PUBMODE: Selects between HWT and DCT as the source of memory transactions.
- **Size:** 1 bit
- **Offset:** 0x20000+0x6e
- **Exists:** Always



**Table 13-248 Fields for Register: PUBMODE**

Bits	Name	Memory Access	Description
0	HwtMemSrc	R/W	<p>HwtMemSrc: When this is set to a 1, the mux that switches between DCT and HWT for the source of memory transactions is switched to HWT.</p> <ul style="list-style-type: none"> <li>■ Setting this bit relinquishes control to the HWT.</li> <li>■ Clearing this bit relinquishes control to the DCT.</li> </ul> <p>The transition can only be made when the FIFOs are all in PtrInit (i.e. all PtrInit signals are high).</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.2.21 DllTrainParam\_pX (for X = 0; X <= 3)

- **Name:** DLL Various Training Parameters
- **Description:** DllTrainParam\_pX: DLL Training Parameters
- **Size:** 8 bits
- **Offset:** 0x20000+0x71+(X\*0x100000)
- **Exists:** Always



**Table 13-249 Fields for Register: DllTrainParam\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
7:4	RxReplicaExtendPhdTime	R/W	<p>RxReplicaExtendPhdTime: Used by the PHY firmware calibrating the LCDL delay cell. Combined with PclkPtrInitVal to set time to sample LCDL CalOut into DfiClk domain. Units if tUI.</p> <p><b>Value After Reset:</b> 0xf</p> <p><b>Exists:</b> Always</p>
3:0	ExtendPhdTIme	R/W	<p>ExtendPhdTIme: Used by the PHY firmware calibrating the LCDL delay cells. Combined with PclkPtrInitVal to set time to sample LCDL CalOut into DfiClk domain. Units if tUI.</p> <p><b>Value After Reset:</b> 0xf</p> <p><b>Exists:</b> Always</p>

### 13.2.22 DllControl

- **Name:** DLL Lock State machine control register
- **Description:** DllControl:  
**Note:**This CSR is written by PHY Initialization Engine (PIE) and the data in here will be overwritten.  
Deprecated
- **Size:** 3 bits
- **Offset:** 0x20000+0x78
- **Exists:** Always



**Table 13-250 Fields for Register: DllControl**

Bits	Name	Memory Access	Description
2	DllResetRSVD	R/W	DllResetRSVD: Deprecated <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
1	DllResetSlave	R/W	DllResetSlave: Deprecated <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
0	DllResetRelock	R/W	DllResetRelock: Deprecated <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.2.23 PulseDIIUpdatePhase

- **Description:** PulseDIIUpdatePhase: Reserved for PHY training firmware use.
- **Size:** 8 bits
- **Offset:** 0x20000+0x79
- **Exists:** Always



**Table 13-251 Fields for Register: PulseDIIUpdatePhase**

Bits	Name	Memory Access	Description
7:0	PulseDIIUpdatePhase	R/W	PulseDIIUpdatePhase: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.2.24 ScratchPadMASTER

- **Description:** ScratchPadMASTER: ScratchPad for PUB\_PAC
- **Size:** 16 bits
- **Offset:** 0x20000+0x7d
- **Exists:** Always

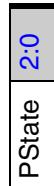


**Table 13-252 Fields for Register: ScratchPadMASTER**

Bits	Name	Memory Access	Description
15:0	ScratchPadMASTER	R/W	<p>ScratchPadMASTER: ScratchPad for PUB_PAC. As required for Automotive. In HMZCAL, When ZcalStopClk=1, this csr can not be accessed.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.2.25 PState

- **Description:** PState: PSTATE Selection
- **Size:** 3 bits
- **Offset:** 0x20000+0x8b
- **Exists:** Always



**Table 13-253 Fields for Register: PState**

Bits	Name	Memory Access	Description
2:0	PState	R/W	<p>PState: This CSR is written by PHY Initialization Engine (PIE) and the data in here will be overwritten. The current PState inferred by the PIE from handshake requests on DFI status interface.</p> <ul style="list-style-type: none"> <li>■ 0 - PState P0</li> <li>■ 1 - PState P1</li> <li>■ 2 - PState P2</li> <li>■ 3 - PState P3</li> <li>■ 7 - PState broadcast</li> </ul> <p>this CSR are used to select the PHY's current Pstate for all Pstateable registers.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.2.26 RxFifoInit

- **Name:** Rx FIFO pointer initialization control
- **Description:** RxFifoInit: This Register enables reset of the dbyte read state
- **Size:** 2 bits
- **Offset:** 0x20000+0xa0
- **Exists:** Always



**Table 13-254 Fields for Register: RxFifoInit**

Bits	Name	Memory Access	Description
1	InhibitRx_fifoRd	R/W	<p>InhibitRx_fifoRd: This field is reserved for training FW use. Setting this inhibits reads of the PHYRXDATAFIFO.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
0	RxFifoInitPtr	R/W	<p>RxFifoInitPtr: When set, the read and write pointers of the dbyte read-data FIFO and of the read credit FIFO, which is the FIFO of the dqsen_sync that extends the read rxen. These state variables may also be reset by update events if csr\$[w]DfiCtrlRxFifoRst=1.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.2.27 ClockingCtrl

- **Description:** ClockingCtrl: This register is dynamically written by PHY Initialization Engine during frequency changes and should not be written by the user.
- **Size:** 2 bits
- **Offset:** 0x20000+0xa2
- **Exists:** Always

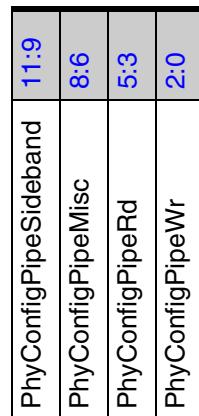


**Table 13-255 Fields for Register: ClockingCtrl**

Bits	Name	Memory Access	Description
1:0	ClockingCtrl	R/W	<p>ClockingCtrl: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

### 13.2.28 PhyPipeConfig

- **Name:** Read Only displays PHY PIPE Configuration.
- **Description:** PhyPipeConfig: Displays PHY PIPEConfiguration
- **Size:** 12 bits
- **Offset:** 0x20000+0xa3
- **Exists:** Always



**Table 13-256 Fields for Register: PhyPipeConfig**

Bits	Name	Memory Access	Description
11:9	PhyConfigPipeSideband	R	<p>PhyConfigPipeSideband: Returns the following value... depending on the chosen define</p> <ul style="list-style-type: none"> <li>■ The defined value if DWC_LPDDR5XPHY_PIPE_DFI_SIDEBAND is defined</li> <li>■ DWC_LPDDR5XPHY_PIPE_DFI_MISC if DWC_LPDDR5XPHY_PIPE_DFI_SIDEBAND is not defined</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always  <b>Reset Mask:</b> 0x0  <b>Volatile:</b> true</p>
8:6	PhyConfigPipeMisc	R	<p>PhyConfigPipeMisc: Returns the following value... depending on the chosen define</p> <ul style="list-style-type: none"> <li>■ The defined value if DWC_LPDDR5XPHY_PIPE_DFI_MISC is defined</li> <li>■ 0x0 if DWC_LPDDR5XPHY_PIPE_DFI_MISC is not defined</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always  <b>Reset Mask:</b> 0x0  <b>Volatile:</b> true</p>

Bits	Name	Memory Access	Description
5:3	PhyConfigPipeRd	R	<p>PhyConfigPipeRd: Returns the following value... depending on the chosen define</p> <ul style="list-style-type: none"> <li>■ The defined value if DWC_LPDDR5XPHY_PIPE_DFI_RD is defined</li> <li>■ 0x0 if DWC_LPDDR5XPHY_PIPE_DFI_RD is not defined</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
2:0	PhyConfigPipeWr	R	<p>PhyConfigPipeWr: Returns the following value... depending on the chosen define</p> <ul style="list-style-type: none"> <li>■ The defined value if DWC_LPDDR5XPHY_PIPE_DFI_WR is defined</li> <li>■ 0x0 if DWC_LPDDR5XPHY_PIPE_DFI_WR is not defined</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.2.29 PhyConfig

- **Name:** Read Only displays PHY Configuration.
- **Description:** PhyConfig: Displays PHY Configuration
- **Size:** 13 bits
- **Offset:** 0x20000+0xa4
- **Exists:** Always

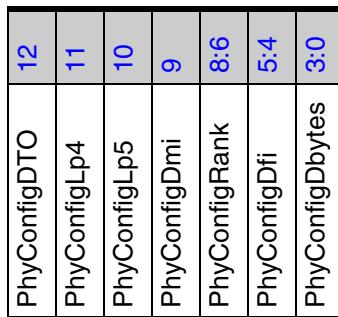


Table 13-257 Fields for Register: PhyConfig

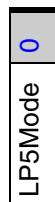
Bits	Name	Memory Access	Description
12	PhyConfigDTO	R	<p>PhyConfigDTO: Returns the following value... depending on the chosen define</p> <ul style="list-style-type: none"> <li>■ 0x1 if DWC_LPDDR5XPHY.DTO_ENABLED is defined</li> <li>■ 0x0 if DWC_LPDDR5XPHY.DTO_ENABLED is not defined</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always  <b>Reset Mask:</b> 0x0  <b>Volatile:</b> true</p>
11	PhyConfigLp4	R	<p>PhyConfigLp4: Reserved, returns 1'b0</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always  <b>Reset Mask:</b> 0x0  <b>Volatile:</b> true</p>

Bits	Name	Memory Access	Description
10	PhyConfigLp5	R	<p>PhyConfigLp5: Returns the following value... depending on the chosen define</p> <ul style="list-style-type: none"> <li>■ 0x1 if DWC_LPDDR5XPHY_LPDDR5_ENABLED is defined</li> <li>■ 0x0 if DWC_LPDDR5XPHY_LPDDR5_ENABLED is not defined</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always  <b>Reset Mask:</b> 0x0  <b>Volatile:</b> true</p>
9	PhyConfigDmi	R	<p>PhyConfigDmi: Returns the following value... depending on the chosen define</p> <ul style="list-style-type: none"> <li>■ 0x1 if DWC_LPDDR5XPHY_DBYTE_DMI_ENABLED is defined</li> <li>■ 0x0 if DWC_LPDDR5XPHY_DBYTE_DMI_ENABLED is not defined</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always  <b>Reset Mask:</b> 0x0  <b>Volatile:</b> true</p>
8:6	PhyConfigRank	R	<p>PhyConfigRank: Returns the following value... depending on the chosen define</p> <ul style="list-style-type: none"> <li>■ 0x1 if DWC_LPDDR5XPHY_NUM_RANKS_1</li> <li>■ 0x2 if DWC_LPDDR5XPHY_NUM_RANKS_2</li> <li>■ 0x4 if DWC_LPDDR5XPHY_NUM_RANKS_4</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always  <b>Reset Mask:</b> 0x0  <b>Volatile:</b> true</p>
5:4	PhyConfigDfi	R	<p>PhyConfigDfi: Returns the following value... depending on the chosen define</p> <ul style="list-style-type: none"> <li>■ 0x1 if DWC_LPDDR5XPHY_NUM_CHANNELS_1</li> <li>■ 0x2 if DWC_LPDDR5XPHY_NUM_CHANNELS_2</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always  <b>Reset Mask:</b> 0x0  <b>Volatile:</b> true</p>

Bits	Name	Memory Access	Description
3:0	PhyConfigDbytes	R	<p>PhyConfigDbytes: Returns the following value... depending on the chosen define</p> <ul style="list-style-type: none"><li>■ 0x2 if DWC_LPDDR5XPHY_NUM_D-BYTES_PER_CHANNEL_2</li></ul> <p><b>Value After Reset:</b> 0x0 <b>Exists:</b> Always <b>Reset Mask:</b> 0x0 <b>Volatile:</b> true</p>

### 13.2.30 LP5Mode

- **Description:** LP5Mode: Selects LP5 Protocol
- **Size:** 1 bit
- **Offset:** 0x20000+0xa5
- **Exists:** Always

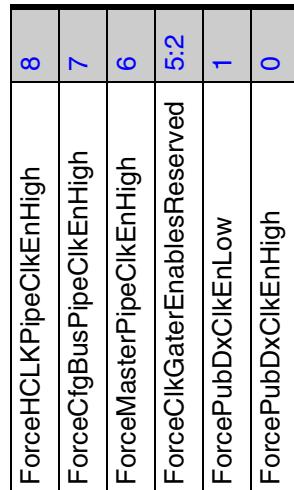


**Table 13-258 Fields for Register: LP5Mode**

Bits	Name	Memory Access	Description
0	LP5Mode	R/W	<p>LP5Mode: When 0, PHY supports LP4X Protocol. When 1, PHY supports LP5 Protocol</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.2.31 ForceClkGaterEnables

- **Name:** Forces internal clock gaters.
- **Description:** ForceClkGaterEnables: Forces internal clock enables High/Low to facilitate test and debug
- **Size:** 9 bits
- **Offset:** 0x20000+0xa6
- **Exists:** Always



**Table 13-259 Fields for Register: ForceClkGaterEnables**

Bits	Name	Memory Access	Description
8	ForceHCLKPipeClkEnHigh	R/W	ForceHCLKPipeClkEnHigh: Forces the HCLK-based PubPiping ClkEn to be 1. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	ForceCfgBusPipeClkEnHigh	R/W	ForceCfgBusPipeClkEnHigh: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6	ForceMasterPipeClkEnHigh	R/W	ForceMasterPipeClkEnHigh: Forces the PUB_PAC Piping ClkEn to be 1. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
5:2	ForceClkGaterEnablesReserved	R/W	ForceClkGaterEnablesReserved: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

Bits	Name	Memory Access	Description
1	ForcePubDxClkEnLow	R/W	<p>ForcePubDxClkEnLow: Forces DfiTxClkEn and DfiRxClkEn to be 0.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
0	ForcePubDxClkEnHigh	R/W	<p>ForcePubDxClkEnHigh: Forces DfiTxClkEn and DfiRxClkEn to be 1.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.2.32 D5Mode

- **Description:** D5Mode: Deprecated
- **Size:** 1 bit
- **Offset:** 0x20000+0xa9
- **Exists:** Always

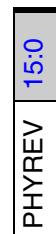


Table 13-260 Fields for Register: D5Mode

Bits	Name	Memory Access	Description
0	D5Mode	R/W	<p>D5Mode: Deprecated</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

### 13.2.33 PHYREV

- **Description:** PHYREV: The hardware version of this PHY, excluding the PUB
- **Size:** 16 bits
- **Offset:** 0x20000+0xee
- **Exists:** Always



**Table 13-261 Fields for Register: PHYREV**

Bits	Name	Memory Access	Description
15:0	PHYREV	R	<p>PHYREV: The hardware version of this PHY, excluding the PUB</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.2.34 TxRdPtrInit

- **Description:** TxRdPtrInit: TxRdPtrInit control register
- **Size:** 1 bit
- **Offset:** 0x20000+0xf8
- **Exists:** Always



**Table 13-262 Fields for Register: TxRdPtrInit**

Bits	Name	Memory Access	Description
0	TxRdPtrInit	R/W	<p>TxRdPtrInit: This register directly controls TxRdPtrInit, and is meant to be written by the PState sequencer as part of the power state switching sequence.</p> <p>It should be written to a 1 once the transition to LP2 is complete and 32 cycles before the de-asserting of init_complete.</p> <p>Likewise it should be written to 0 once the transition to an active PState is complete and 32 cycles before the assertion of init_complete.</p> <p>It should reset to a 1 on both warm and cold reset.</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

### 13.2.35 MASTERReservedX (for X == 0)

- **Description:** MASTERReservedX: Reserved for future use
- **Size:** 16 bits
- **Offset:** 0x20000+0xfb
- **Exists:** Always



**Table 13-263 Fields for Register: MASTERReservedX (for X == 0)**

Bits	Name	Memory Access	Description
15:0	MASTERReserved0	R/W	<p>MASTERReserved0: Reserved for future use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.2.36 PUBReservedP1\_pX (MASTER) (for X = 0; X <= 3)

- **Description:** PUBReservedP1\_pX: Reserved for future use
- **Size:** 8 bits
- **Offset:** 0x20000+0xff+(X\*0x100000)
- **Exists:** Always

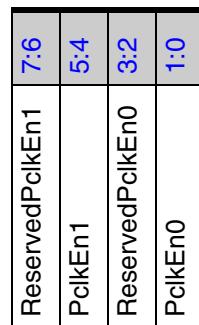


**Table 13-264 Fields for Register: PUBReservedP1\_pX (MASTER) (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
7:0	PUBReservedP1_p0	R/W	PUBReservedP1_p0: Reserved for future use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.2.37 PclkGateControl

- **Name:** Pclk gating override control
- **Description:** PclkGateControl: Clock gating override control for all Pclk\* clock signals going out of the PAC hard macro  
**Note:** This CSR is written by DevInit and PHY Initialization Engine (PIE) and should not be written by the user.
- **Size:** 8 bits
- **Offset:** 0x20000+0x200
- **Exists:** Always



**Table 13-265 Fields for Register: PclkGateControl**

Bits	Name	Memory Access	Description
7:6	ReservedPclkEn1	R/W	ReservedPclkEn1: Reserved for future. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
5:4	PclkEn1	R/W	PclkEn1: Clock gating override control for channel 1 Pclks <ul style="list-style-type: none"> <li>■ 2'b0x - Pclk*1 clocks gating is controlled by PUB_PAC pipeline</li> <li>■ 2'b10 - Gate all Pclk*1 clock outputs from the PAC hard macro</li> <li>■ 2'b11 - Enable all Pclk*1 clock outputs from the PAC hard macro, with the divide ratios specified in PclkDiv*1 CSRs</li> </ul> <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
3:2	ReservedPclkEn0	R/W	ReservedPclkEn0: Reserved for future. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

Bits	Name	Memory Access	Description
1:0	PclkEn0	R/W	<p>PclkEn0: Clock gating override control for channel 0 Pclks</p> <ul style="list-style-type: none"> <li>■ 2'b0x - Pclk*0 clocks gating is controlled by PUB_PAC pipeline</li> <li>■ 2'b10 - Gate all Pclk*0 clock outputs from the PAC hard macro</li> <li>■ 2'b11 - Enable all Pclk*0 clock outputs from the PAC hard macro, with the divide ratios specified in PclkDiv*0 CSRs</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

## 13.3 **DWC\_DDRPHYA\_ACj\_Pk Registers**

### 13.3.1 AcPipeEn\_pX (for X = 0; X <= 3)

- **Description:** AcPipeEn\_pX: Enable DfiClk Pipeline on the Command Address.
- **Size:** 2 bits
- **Offset:**  $(0x30000 + (j \ll 12)) + 0x8 + (X * 0x100000)$
- **Exists:** Always



**Table 13-266 Fields for Register: AcPipeEn\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
1:0	AcPipeEn_p0	R/W	<p>AcPipeEn_p0: This field is programmed as follows:</p> <ul style="list-style-type: none"> <li>■ When set to 1, Design add 1 DfiClk of extra latency on the Command Address.</li> <li>■ When set to 2, Design add 2 DfiClk of extra latency on the Command Address.</li> <li>■ Write and Read DFI Timing parameter will have to be adjusted accordingly.</li> </ul> <p>This is useful to enable to support lower frequency setting in LPDDR5 Mode.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.2 CKDIIStopCal

- **Description:** CKDIIStopCal: Stop MDLL Calibration for incoming TxRdPtrInit
- **Size:** 1 bit
- **Offset:** (0x30000+(j<<12))+0xa
- **Exists:** Always



**Table 13-267 Fields for Register: CKDIIStopCal**

Bits	Name	Memory Access	Description
0	CKDIIStopCal	R/W	<p>CKDIIStopCal: Stops the MDLL LCDL 16 DfiClk cycles before the TxRdPtrInit asserts. to ensure no interrupted calibrating data is sent out. 0, the MDLL calibration is active. 1, the MDLL calibration is stopped.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.3 CkVal\_pX (for X = 0; X <= 3)

- **Description:** CkVal\_pX: Active clock encoding register for MEMCLK
- **Size:** 2 bits
- **Offset:** (0x30000+(j<<12))+0xe+(X\*0x100000)
- **Exists:** Always



**Table 13-268 Fields for Register: CkVal\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
1:0	CkVal_p0	R/W	<p>CkVal_p0: This register is used to specify the running MEMCLK state. bit 0 corresponds to CK_c, and bit 1 corresponds to CK_t</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

### 13.3.4 CkDisVal\_pX (for X = 0; X <= 3)

- **Description:** CkDisVal\_pX: Mode select register for MEMCLK
- **Size:** 2 bits
- **Offset:** (0x30000+(j<<12))+0xf+(X\*0x100000)
- **Exists:** Always



**Table 13-269 Fields for Register: CkDisVal\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
1	CkDisVal_p0	R/W	<p>CkDisVal_p0: This register is used to specify the stopped MEMCLK state. The PHY provides 1 memory clocks per channel</p> <ul style="list-style-type: none"> <li>■ When the toggling of memory clock CK_t is disabled with dfi_clk_disable=1, the memory clock CK_t is driven with Register CkDisVal[1].</li> </ul> <p><b>Note:</b>In LPDDR4X/5 CK_t and CK_c needs to be always stopped inverted.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
0	Reserved	R	<p>Reserved: Reserved</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.3.5 DTOBypassEn

- **Description:** DTOBypassEn: Async Flyover Tx Mode control for DTO
- **Size:** 1 bit
- **Offset:** (0x30000+(j<<12))+0x10
- **Exists:** Always



**Table 13-270 Fields for Register: DTOBypassEn**

Bits	Name	Memory Access	Description
0	DTOBypassEn	R/W	<p>DTOBypassEn: When set, overrides MTEST output of BP_DTO with MTestData</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.6 ACSingleEndedMode\_pX (for X = 0; X <= 3)

- **Name:** Control for Single Ended LP4X/5 Mode
- **Description:** ACSingleEndedMode\_pX: Configures the DIFF IO to operate in Single Ended Mode.
- **Size:** 1 bit
- **Offset:** (0x30000+(j<<12))+0x15+(X\*0x100000)
- **Exists:** Always



**Table 13-271 Fields for Register: ACSingleEndedMode\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
0	SingleEndedCK	R/W	<p>SingleEndedCK: Drives TxDataC of CK DIFF Slice to 0. This is applicable in LPDDR4X and LPDDR5 Mode.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.7 InitSeqControl

- **Name:** AC/DBYTE Pipeline Init Sequence/Pipeline Control register
- **Description:** InitSeqControl: This register directly controls whether the overall AC/DBYTE pipeline initialization sequence steps can be automatically triggered by TxRdPtrInit pipeline, or only controlled/set by PState sequencer by writing manually to each independent init control CSR below. The PState sequencer must write to each independent init control, if the TxPdPtrInit Pipeline is disabled. The PState sequencer must satisfy all the rules & restriction of the init control sequence as per describe in Arch spec - Ex. TxRdPtrInit=1 --> TxRdPtrInit=0 --> CMDFIFO running --> other init control. This is to allow the PState sequencer to have full controllability of the init sequence if needed.
- **Size:** 16 bits
- **Offset:** (0x30000+(j<<12))+0x16
- **Exists:** Always

ReservedInitSeqControl	15
InitControlRxReplSeqInit	14
InhibitTxRdPtrRxRepSeqInit	13
InitControlDbRxEnPhUpdInit	12
InitControlDbPptInit	11
InhibitTxRdPtrDbPptInit	10
InitControlDbDataPipeInit	9
InhibitTxRdPtrDbRxEnPhUpdInit	8
InhibitTxRdPtrDbDataPipeInit	7
InitControlTxXFIFOInit	6
InhibitTxRdPtrTXXFIFOInit	5
InitControlRxReplLcdlInit	4
InhibitTxRdPtrRxRepLcdlInit	3
InitControlRstLclCal	2
InhibitTxRdPtrRstLclCal	1
InhibitTxRdPtrBypassForce	0

Table 13-272 Fields for Register: InitSeqControl

Bits	Name	Memory Access	Description
15	ReservedInitSeqControl	R/W	ReservedInitSeqControl: Reserved Bits <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
14	InitControlRxReplSeqInit	R/W	InitControlRxReplSeqInit: When set (1), this sets the RxReplica calibration sequencer in Init State. This is not used by PUB AC block. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

Bits	Name	Memory Access	Description
13	InhibitTxRdPtrRxRepSeqInit	R/W	<p>InhibitTxRdPtrRxRepSeqInit: Inhibit the TxRdPtrInit Pipeline from resetting the RxReplica Sequencer When set (1), the PState sequencer may reset the LCDL calibration registers via InitControlRxRepSeqInit, in absence of the Pipeline Control This is not used by PUB AC block.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
12	InitControlDbRxEnPhUpdInit	R/W	<p>InitControlDbRxEnPhUpdInit: Whenever toggle (0-&gt;1), this creates a RxEnPhaseUpdate pulse to reset RxEnPhaseA/B logic Note that this does not changes/reset any RxEn Phase Delay selection/value This is not used by PUB AC block.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
11	InitControlDbPptInit	R/W	<p>InitControlDbPptInit: When set (1), this resets the DByte PPT block This is not used by PUB AC block.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
10	InhibitTxRdPtrDbPptInit	R/W	<p>InhibitTxRdPtrDbPptInit: Inhibit the TxRdPtrInit Pipeline from resetting the DByte PPT block. When set (1), The PState sequencer may reset the DByte PPT block via csrInitControlDbPptInit, in absence of the Pipeline Control This is not used by PUB AC block.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
9	InitControlDbDataPipeInit	R/W	<p>InitControlDbDataPipeInit: When set (1), this resets the internal data-pipe registers This is not used by PUB AC block.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
8	InhibitTxRdPtrDbRxEnPhUpdInit	R/W	<p>InhibitTxRdPtrDbRxEnPhUpdInit: Inhibit the TxRdPtrInit Pipeline from creating a RxEnPhaseUpdate pulse to reset RxEnPhaseA/B logic Note that this does not changes/reset any RxEn Phase Delay selection/value When set (1), The PState sequencer may reset the internal data-pipe registers via csrInitControlDbRxEnPhUpdInit, in absence of the Pipeline Control This is not used by PUB AC block.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
7	InhibitTxRdPtrDbDataPipeInit	R/W	<p>InhibitTxRdPtrDbDataPipeInit: Inhibit the TxRdPtrInit Pipeline from resetting the internal data-pipe registers When set (1), The PState sequencer may reset the internal data-pipe registers via csrInitControlDbDataPipeInit, in absence of the Pipeline Control This is not used by PUB AC block.</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
6	InitControlTXFIFOInit	R/W	<p>InitControlTXFIFOInit: Whenever toggle (0-&gt;1), this trigger 1 round of TXFIFO (DATA FIFO) content initialization cycle This min high-low pulse width of this InitControlTXFIFOInit must be min 26 DfiPubClk cycles</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
5	InhibitTxRdPtrTXFIFOInit	R/W	<p>InhibitTxRdPtrTXFIFOInit: Inhibit the TxRdPtrInit Pipeline from initializing the TXFIFO (DATA FIFO) content initialization When set (1), the PState sequencer may initialize the RxReplica Sequencer via csrInitControlTXFIFOInit, in absence of the Pipeline Control</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
4	InitControlRxReplLcdlInit	R/W	<p>InitControlRxReplLcdlInit: When set (1), this resets the RxReplica LCDL calibration registers (Pclk domain) This is not used by PUB AC block.</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
3	InhibitTxRdPtrRxReplLcdlInit	R/W	<p>InhibitTxRdPtrRxReplLcdlInit: Inhibit the TxRdPtrInit Pipeline from resetting the RxReplica LCDL calibration registers (Pclk domain) When set (1), the PState sequencer may reset the LCDL calibration registers via InitControlRxReplLcdlInit, in absence of the Pipeline Control This is not used by PUB AC block.</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
2	InitControlRstLclCal	R/W	<p>InitControlRstLclCal: When set (1), this resets the LCDL calibration registers (Pclk domain)</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
1	InhibitTxRdPtrRstLclCal	R/W	<p>InhibitTxRdPtrRstLclCal: Inhibit the TxRdPtrInit Pipeline from resetting the LCDL calibration registers (Pclk domain). When set (1), the PState sequencer may reset the LCDL calibration registers via csrlInitControlRstLclCal, in absence of the Pipeline Control</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
0	InhibitTxRdPtrBypassForce	R/W	<p>InhibitTxRdPtrBypassForce: Inhibit the TxRdPtrInit Pipeline from forcing the IO into Bypass Mode. When set (1), the PState sequencer may control the IO Bypass via csrAsync*TxMode, in absence of the Pipeline Control</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.8 MtestMuxSel

- **Description:** MtestMuxSel: Digital Observation Pin control
- **Size:** 10 bits
- **Offset:** (0x30000+(j<<12))+0x1a
- **Exists:** Always



**Table 13-273 Fields for Register: MtestMuxSel**

Bits	Name	Memory Access	Description
9:0	MtestMuxSel	R/W	<p>MtestMuxSel: Controls for the mux for asynchronous data to the Digital Observation Pin.</p> <ul style="list-style-type: none"> <li>■ Encoding 9'h0 causes this chiplet to drive 0, (allowing flat 'OR' of pass-through information).</li> <li>■ MtestMuxSel[4:0] - Lower 5 bits selects one bit from the 32 bit MtestMux in each section or slice.</li> </ul> <p>Detailed tables are in the PUB Databook for PUB sections (AC, MASTER, etc.)</p> <ul style="list-style-type: none"> <li>■ MtestMuxSel[8:5] - Where more than one MtestMux exists, non-zero values select the outputs of the additional Mtest-Muxes,           <ul style="list-style-type: none"> <li>□ DBYTE MtestMuxSel[6:5]=2'h0 for Mux-A and 2'h1 for Mux-B and 2'h2 for Mux-C.</li> <li>□ For all other slaves, MtestMuxSel[8:5] are unused</li> </ul> </li> </ul> <p><b>Note:</b> See the PUB Databook for how, or if, the Digital Observation Pin is mapped to a physical bump in this configuration.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.9 PorControl

- **Name:** PMU Power-on Reset Control
- **Description:** PorControl: Power On Reset Control. Register is written PIE during PHY operation.
- **Size:** 1 bit
- **Offset:**  $(0x30000 + (j \ll 12)) + 0x20$
- **Exists:** Always



**Table 13-274 Fields for Register: PorControl**

Bits	Name	Memory Access	Description
0	PwrOkDlyCtrl	R/W	<p>PwrOkDlyCtrl: Set in Phyinit step C, PIE to 1 during execution, cleared on PHY Reset or on power cycle. Used to delay PwrOk_VDD, PwrOk_VDDQ and PwrOk_VAA assertion, after power ramp up. When 0, forces PwrOk_VDD, PwrOk_VDDQ and PwrOk_VAA to 0.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.10 ClrPORMemReset

- **Description:** ClrPORMemReset: Switch the POR Self-Initialized LATCH to 1
- **Size:** 1 bit
- **Offset:** (0x30000+(j<<12))+0x21
- **Exists:** Always



**Table 13-275 Fields for Register: ClrPORMemReset**

Bits	Name	Memory Access	Description
0	ClrPORMemReset	R/W	<p>ClrPORMemReset: Used to switch the POR Self-Initialized LATCH state so that MemReset_ForceData changes to 1.</p> <p><b>Note:</b> This CSR is written by PHY Initialization Engine (PIE) and DevInit and the data in here will be overwritten.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.11 MemResetL

- **Name:** Protection and control of BP\_MEMRESET\_L
- **Description:** MemResetL: Control the BP\_MEMRESET\_L output of the PHY. Intended to be used to protect/force the BP\_MEMRESET\_L output.
- **Size:** 3 bits
- **Offset:** (0x30000+(j<<12))+0x22
- **Exists:** Always



**Table 13-276 Fields for Register: MemResetL**

Bits	Name	Memory Access	Description
2	AsyncMemResetL RxMode	R/W	<p>AsyncMemResetL RxMode: Used for giving firmware direct access to BP_MEMRESET_L pin</p> <ul style="list-style-type: none"> <li>■ When this bit is set high, BP_MEMRESET_L flyover Receiver gets enabled</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
1	ProtectMemReset	R/W	<p>ProtectMemReset: Control the BP_MEMRESET_L output of the PHY. The state of this field is transferred to the PHY BP_MEMRESET_L output when ProtectMemReset is set.</p> <p><b>Value After Reset:</b> 0x1  <b>Exists:</b> Always</p>
0	MemResetL Value	R/W	<p>MemResetL Value: Control the BP_MEMRESET_L output of the PHY. The state of this field is transferred to the PHY BP_MEMRESET_L output when ProtectMemReset is set.</p> <p><b>Value After Reset:</b> 0x1  <b>Exists:</b> Always</p>

### 13.3.12 CMOSxHardMacroModeSel

- **Description:** CMOSxHardMacroModeSel: This CSR is deprecated
- **Size:** 1 bit
- **Offset:** (0x30000+(j<<12))+0x24
- **Exists:** Always



**Table 13-277 Fields for Register: CMOSxHardMacroModeSel**

Bits	Name	Memory Access	Description
0	CMOSxHardMacroModeSel	R/W	<p>CMOSxHardMacroModeSel: Deprecated</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.13 RxAcVrefControl\_pX (for X = 0; X <= 3)

- **Name:** RxAcVref DAC control.
- **Description:** RxAcVrefControl\_pX: This register controls the RxAcVref DAC to generate VREF for receiver AC pins
- **Size:** 9 bits
- **Offset:** (0x30000+(j<<12))+0x25+(X\*0x100000)
- **Exists:** Always



**Table 13-278 Fields for Register: RxAcVrefControl\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
8	RxAcVrefDacEn	R/W	<p>RxAcVrefDacEn: When asserted, VREFAC value is applied to the AC slice bypass mode receiver</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
7:0	RxAcVrefDac	R/W	<p>RxAcVrefDac: The RxAcVref DAC Code signals and control the DAC voltage output Max allowed value is binary 11001011 (decimal 203) DAC voltage (V) = VDD2H (V) * RxAcVrefDac[7:0] / 256</p> <ul style="list-style-type: none"> <li>■ 8'b00100000 - Default</li> </ul> <p>Refer to Technology specific PHY databook Custom CKT Macro Optimal Settings for any recommended settings, as available</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.14 MemResetLStatus

- Name:** This Register provides BP\_MEMRESET\_L pad status
- Description:** MemResetLStatus: This register provides status bits for POR custom circuit value and BP\_MEMRESET\_L Async Flyover RxData
- Size:** 2 bits
- Offset:** (0x30000+(j<<12))+0x26
- Exists:** Always



Table 13-279 Fields for Register: MemResetLStatus

Bits	Name	Memory Access	Description
1	AsyncMemResetLRxData	R	<p>AsyncMemResetLRxData: RxData bits for BP_MEMRESET_L Async Flyover</p> <ul style="list-style-type: none"> <li>Used for giving firmware direct access to BP_MEMRESET_L Rx Data</li> <li>When AsyncMemResetRxMode is set, this bit provides BP_MEMRESET_L flyover Rx Data</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always  <b>Reset Mask:</b> 0x0  <b>Volatile:</b> true</p>
0	PORMemReset	R	<p>PORMemReset: Returns the POR custom circuit value which drives the memory RESET signal, when PwrOk_VAA = 1'b0</p> <ul style="list-style-type: none"> <li>This signal powers on 1'b0.</li> <li>This signal goes 1'b1 during the PHY Initialization sequence.</li> <li>This signal goes 1'b0 when VAA &lt; 0.3V.</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always  <b>Reset Mask:</b> 0x0  <b>Volatile:</b> true</p>

### 13.3.15 ACDlyScaleGatingDisable

- **Description:** ACDlyScaleGatingDisable: Reserved for PHY training firmware use.
- **Size:** 1 bit
- **Offset:** (0x30000+(j<<12))+0x27
- **Exists:** Always



**Table 13-280 Fields for Register: ACDlyScaleGatingDisable**

Bits	Name	Memory Access	Description
0	ACDlyScaleGatingDisable	R/W	ACDlyScaleGatingDisable: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always

### 13.3.16 LcdlCalControl

- **Name:** DLL Lock State machine control register
- **Description:** LcdlCalControl:  
**Note:**This CSR is written by PHY Initialization Engine (PIE) and the data in here will be overwritten.
- **Size:** 3 bits
- **Offset:** (0x30000+(j<<12))+0x47
- **Exists:** Always



**Table 13-281 Fields for Register: LcdlCalControl**

Bits	Name	Memory Access	Description
2	LcdlUpdTrackDis	R/W	<p>LcdlUpdTrackDis: Disables LCDL calibration sequence on update events. Typically not used.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
1	LcdlCalStop	R/W	<p>LcdlCalStop: Pulsing high stops the locking sequence.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
0	LcdlCalResetRelock	R/W	<p>LcdlCalResetRelock: Pulsing high changes the LCDL state to unlocked. If the macro command FIFOs are not running, as when they are held in pointer init, then a seeded relock sequence begins immediately after pointer init when the command FIFOs are running. If the macro command FIFOs are running, then a seeded relock sequence begins immediately; this is not to be done usually.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.17 ACParityInvert

- **Description:** ACParityInvert: Invert APB Parity for register slave AC
- **Size:** 2 bits
- **Offset:** (0x30000+(j<<12))+0x4d
- **Exists:** Always



**Table 13-282 Fields for Register: ACParityInvert**

Bits	Name	Memory Access	Description
1:0	ACParityInvert	R/W	<p>ACParityInvert: Invert APB Parity for register slave AC. As required for Automotive. NOTE: This register should be used only for test. Set the bits for only one slave at a time. When bits are set for a particular slave. APB Reads of only that slave are valid. Bit 0 applies to [7:0] Bit 1 applies to [15:8] In HMZCAL, When ZcalStopClk=1, this csr can not be accessed.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.18 ACPulseDIIUpdatePhase

- **Description:** ACPulseDIIUpdatePhase: Reserved for PHY training firmware use.
- **Size:** 3 bits
- **Offset:** (0x30000+(j<<12))+0x79
- **Exists:** Always



**Table 13-283 Fields for Register: ACPulseDIIUpdatePhase**

Bits	Name	Memory Access	Description
2:0	ACPulseDIIUpdatePhase	R/W	<p>ACPulseDIIUpdatePhase: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.19 ScratchPadAC

- **Description:** ScratchPadAC: ScratchPad for AC
- **Size:** 16 bits
- **Offset:**  $(0x30000 + (j << 12)) + 0x7d$
- **Exists:** Always

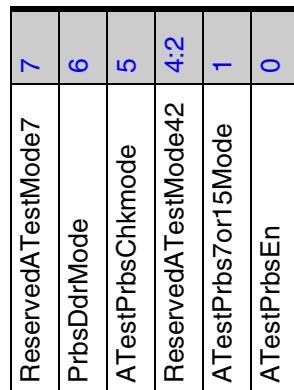


**Table 13-284 Fields for Register: ScratchPadAC**

Bits	Name	Memory Access	Description
15:0	ScratchPadAC	R/W	<p>ScratchPadAC: ScratchPad for AC. As required for Automotive. In HMZCAL, When ZcalStopClk=1, this csr can not be accessed.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.20 ATestMode

- **Name:** ATestMode control for SE/Diff Address/Command slices
- **Description:** ATestMode: Controls for Address Command Loopback. This CSR is written by ATE firmware.
- **Size:** 8 bits
- **Offset:** (0x30000+(j<<12))+0x7f
- **Exists:** Always



**Table 13-285 Fields for Register: ATestMode**

Bits	Name	Memory Access	Description
7	ReservedATestMode7	R/W	ReservedATestMode7: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6	PrbsDdrMode	R/W	PrbsDdrMode: Set this bit to run AC loopback in DDR mode. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
5	ATestPrbsChkmode	R/W	ATestPrbsChkmode: Arms the loopback PRBS7/PRBS15 checker of all the DDR output pins in this chiplet. This bit must be set at least 8 DfiClks after setting ATestPrbsEn to allow proper seeding <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
4:2	ReservedATestMode42	R/W	ReservedATestMode42: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

Bits	Name	Memory Access	Description
1	ATestPrbs7or15Mode	R/W	<p>ATestPrbs7or15Mode: Select between prbs7 and prbs15 polynomial for prbs checker.</p> <ul style="list-style-type: none"> <li>■ 0 selects prbs7 polynomial for self seeding checker</li> <li>■ 1 selects prbs15 polynomial for self seeding checker</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
0	ATestPrbsEn	R/W	<p>ATestPrbsEn: Enables seeding for loopback PRBS7/PRBS15 testing of all the DDR output pins in this chiplet.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.21 AcDigStrobeGenSel

- **Description:** AcDigStrobeGenSel: Selects the source for the generation of the RxDigStrobe
- **Size:** 1 bit
- **Offset:** (0x30000+(j<<12))+0x86
- **Exists:** Always



**Table 13-286 Fields for Register: AcDigStrobeGenSel**

Bits	Name	Memory Access	Description
0	AcDigStrobeGenSel	R/W	<p>AcDigStrobeGenSel: This field is programmed as follows:</p> <ul style="list-style-type: none"> <li>■ 1'b1 - RxDigStrobe is Generated from CSR.</li> <li>■ 1'b0 - RxDigStrobe is Generated from Read Transaction.</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.22 AcDigStrobePat

- **Description:** AcDigStrobePat: Selects the pattern for RxDigStrobe
- **Size:** 4 bits
- **Offset:** (0x30000+(j<<12))+0x87
- **Exists:** Always

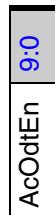


**Table 13-287 Fields for Register: AcDigStrobePat**

Bits	Name	Memory Access	Description
3:0	AcDigStrobePat	R/W	<p>AcDigStrobePat: 4 UI Pattern for the generation RxDigStrobe. This pattern will be set on the RxDigStrobe if AcDigStrobeGenSel = 1'b1</p> <p><b>Value After Reset:</b> 0xa</p> <p><b>Exists:</b> Always</p>

### 13.3.23 AcOdtEn

- **Description:** AcOdtEn: Per Slice in AC OdtEn Control
- **Size:** 10 bits
- **Offset:** (0x30000+(j<<12))+0x88
- **Exists:** Always



**Table 13-288 Fields for Register: AcOdtEn**

Bits	Name	Memory Access	Description
9:0	AcOdtEn	R/W	<p>AcOdtEn: This field is programmed as follows: Lanes: LP5 Lanes: Lower Upper</p> <ul style="list-style-type: none"> <li>■ 0 Tx SE lane0 BP[0] BP[9]</li> <li>■ 1 Tx SE lane1 BP[1] BP[10]</li> <li>■ 2 Tx SE lane2 BP[2] BP[11]</li> <li>■ 3 Tx SE lane3 BP[3] BP[12]</li> <li>■ 4 Tx SE lane4 BP[6] BP[15]</li> <li>■ 5 Tx SE lane5 BP[7] BP[16]</li> <li>■ 6 Tx SE lane6 BP[8] BP[17]</li> <li>■ 7 Tx SE lane7 BP.DTO BP[17]</li> <li>■ 8 Tx SEC lane0 BP[4] BP[13]</li> <li>■ 9 Tx SEC lane1 BP[5] BP[14]</li> <li>■ 10 Tx DIFF0 lane</li> <li>■ 11 Unused</li> <li>■ 12 DTO</li> </ul> <p><b>Value After Reset:</b> 0x0 <b>Exists:</b> Always</p>

### 13.3.24 AcRxStrobeEnPat

- **Description:** AcRxStrobeEnPat: Selects the pattern for RxStrobeEn for AC DIFF Slice.
- **Size:** 4 bits
- **Offset:** (0x30000+(j<<12))+0x89
- **Exists:** Always

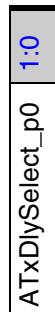


**Table 13-289 Fields for Register: AcRxStrobeEnPat**

Bits	Name	Memory Access	Description
3:0	AcRxStrobeEnPat	R/W	AcRxStrobeEnPat: 4 UI Pattern for the generation RxStrobeEn <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.3.25 ATxDlySelect\_pX (for X = 0; X <= 3)

- **Description:** ATxDlySelect\_pX: Address/Command Delay Select
- **Size:** 2 bits
- **Offset:** (0x30000+(j<<12))+0x8f+(X\*0x100000)
- **Exists:** Always



**Table 13-290 Fields for Register: ATxDlySelect\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
1:0	ATxDlySelect_p0	R/W	<p>ATxDlySelect_p0: Selects between csrAt[xlu]Dly and csrAt[xlu]Dly2nMode as determined by dfi_2n_mode.</p> <ul style="list-style-type: none"> <li>■ 0x: Select csrAC[X]TxDly when !dfi_2n_mode. Otherwise select csrAC[X]TxDly2nMode</li> <li>■ 10: Select csrAC[X]TxDly</li> <li>■ 11: Select csrAC[X]TxDly2nMode</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.26 MapCA0toDfi

- **Description:** MapCA0toDfi: Maps AC CH PHY CA lane 0 from dfi\_address
- **Size:** 4 bits
- **Offset:**  $(0x30000 + (j \ll 12)) + 0x90$
- **Exists:** Always



**Table 13-291 Fields for Register: MapCA0toDfi**

Bits	Name	Memory Access	Description
3:0	MapCA0toDfi	R/W	<p>MapCA0toDfi: These CSRs map a dfi_address to CA 0 of corresponding AC channel. Each register in the set of MapCA0toDfi must have a unique value within the set. For example, if PHY CA0 is mapped from dfi_address[2], then Register MapCA0toDfi should be 2.</p> <ul style="list-style-type: none"> <li>■ In LP4X registers MapCA[0..5]toDfi are used, values 0-5 are valid, values 6 is reserved.</li> <li>■ In LP5 registers MapCA[0..6]toDfi are used.</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.27 MapCA1toDfi

- **Description:** MapCA1toDfi: Maps AC CH PHY CA lane 1 from dfi\_address
- **Size:** 4 bits
- **Offset:**  $(0x30000 + (j \ll 12)) + 0x91$
- **Exists:** Always



**Table 13-292 Fields for Register: MapCA1toDfi**

Bits	Name	Memory Access	Description
3:0	MapCA1toDfi	R/W	<p>MapCA1toDfi: These CSRs map a dfi_address to CA 1 of corresponding AC channel. Each register in the set of MapCA1toDfi must have a unique value within the set. For example, if PHY CA1 is mapped from dfi_address[2], then Register MapCA1toDfi should be 2.</p> <ul style="list-style-type: none"> <li>■ In LP4X registers MapCA[0..5]toDfi are used, values 0-5 are valid, values 6 is reserved.</li> <li>■ In LP5 registers MapCA[0..6]toDfi are used.</li> </ul> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

### 13.3.28 MapCA2toDfi

- **Description:** MapCA2toDfi: Maps AC CH PHY CA lane 2 from dfi\_address
- **Size:** 4 bits
- **Offset:**  $(0x30000 + (j \ll 12)) + 0x92$
- **Exists:** Always



**Table 13-293 Fields for Register: MapCA2toDfi**

Bits	Name	Memory Access	Description
3:0	MapCA2toDfi	R/W	<p>MapCA2toDfi: These CSRs map a dfi_address to CA 2 of corresponding AC channel. Each register in the set of MapCAtoDfi must have a unique value within the set. For example, if PHY CA2 is mapped from dfi_address[2], then Register MapCA2toDfi should be 2.</p> <ul style="list-style-type: none"> <li>■ In LP4X registers MapCA[0..5]toDfi are used, values 0-5 are valid, values 6 is reserved.</li> <li>■ In LP5 registers MapCA[0..6]toDfi are used.</li> </ul> <p><b>Value After Reset:</b> 0x2</p> <p><b>Exists:</b> Always</p>

### 13.3.29 MapCA3toDfi

- **Description:** MapCA3toDfi: Maps AC CH PHY CA lane 3 from dfi\_address
- **Size:** 4 bits
- **Offset:**  $(0x30000 + (j \ll 12)) + 0x93$
- **Exists:** Always



**Table 13-294 Fields for Register: MapCA3toDfi**

Bits	Name	Memory Access	Description
3:0	MapCA3toDfi	R/W	<p>MapCA3toDfi: These CSRs map a dfi_address to CA 3 of corresponding AC channel. Each register in the set of MapCAtoDfi must have a unique value within the set. For example, if PHY CA3 is mapped from dfi_address[2], then Register MapCA3toDfi should be 2.</p> <ul style="list-style-type: none"> <li>■ In LP4X registers MapCA[0..5]toDfi are used, values 0-5 are valid, values 6 is reserved.</li> <li>■ In LP5 registers MapCA[0..6]toDfi are used.</li> </ul> <p><b>Value After Reset:</b> 0x3</p> <p><b>Exists:</b> Always</p>

### 13.3.30 MapCA4toDfi

- **Description:** MapCA4toDfi: Maps AC CH PHY CA lane 4 from dfi\_address
- **Size:** 4 bits
- **Offset:**  $(0x30000 + (j \ll 12)) + 0x94$
- **Exists:** Always



**Table 13-295 Fields for Register: MapCA4toDfi**

Bits	Name	Memory Access	Description
3:0	MapCA4toDfi	R/W	<p>MapCA4toDfi: These CSRs map a dfi_address to CA 4 of corresponding AC channel. Each register in the set of MapCAtoDfi must have a unique value within the set. For example, if PHY CA4 is mapped from dfi_address[2], then Register MapCA4toDfi should be 2.</p> <ul style="list-style-type: none"> <li>■ In LP4X registers MapCA[0..5]toDfi are used, values 0-5 are valid, values 6 is reserved.</li> <li>■ In LP5 registers MapCA[0..6]toDfi are used.</li> </ul> <p><b>Value After Reset:</b> 0x4</p> <p><b>Exists:</b> Always</p>

### 13.3.31 MapCA5toDfi

- **Description:** MapCA5toDfi: Maps AC CH PHY CA lane 5 from dfi\_address
- **Size:** 4 bits
- **Offset:** (0x30000+(j<<12))+0x95
- **Exists:** Always



**Table 13-296 Fields for Register: MapCA5toDfi**

Bits	Name	Memory Access	Description
3:0	MapCA5toDfi	R/W	<p>MapCA5toDfi: These CSRs map a dfi_address to CA 5 of corresponding AC channel. Each register in the set of MapCAtoDfi must have a unique value within the set. For example, if PHY CA5 is mapped from dfi_address[2], then Register MapCA5toDfi should be 2.</p> <ul style="list-style-type: none"> <li>■ In LP4X registers MapCA[0..5]toDfi are used, values 0-5 are valid, values 6 is reserved.</li> <li>■ In LP5 registers MapCA[0..6]toDfi are used.</li> </ul> <p><b>Value After Reset:</b> 0x5</p> <p><b>Exists:</b> Always</p>

### 13.3.32 MapCA6toDfi

- **Description:** MapCA6toDfi: Maps AC CH PHY CA lane 6 from dfi\_address
- **Size:** 4 bits
- **Offset:**  $(0x30000 + (j \ll 12)) + 0x96$
- **Exists:** Always



**Table 13-297 Fields for Register: MapCA6toDfi**

Bits	Name	Memory Access	Description
3:0	MapCA6toDfi	R/W	<p>MapCA6toDfi: These CSRs map a dfi_address to CA 6 of corresponding AC channel. Each register in the set of MapCA6toDfi must have a unique value within the set. For example, if PHY CA6 is mapped from dfi_address[2], then Register MapCA6toDfi should be 2.</p> <ul style="list-style-type: none"> <li>■ In LP4X registers MapCA[0..5]toDfi are used, values 0-5 are valid, values 6 is reserved.</li> <li>■ In LP5 registers MapCA[0..6]toDfi are used.</li> </ul> <p><b>Value After Reset:</b> 0x6</p> <p><b>Exists:</b> Always</p>

### 13.3.33 AsyncAcTxMode

- **Description:** AsyncAcTxMode: Reserved for PHY training firmware use.
- **Size:** 14 bits
- **Offset:** (0x30000+(j<<12))+0xa0
- **Exists:** Always



**Table 13-298 Fields for Register: AsyncAcTxMode**

Bits	Name	Memory Access	Description
13:0	AsyncAcTxMode	R/W	<p>AsyncAcTxMode: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.34 AsyncAcRxMode

- **Description:** AsyncAcRxMode: Reserved for PHY training firmware use.
- **Size:** 14 bits
- **Offset:** (0x30000+(j<<12))+0xa1
- **Exists:** Always



**Table 13-299 Fields for Register: AsyncAcRxMode**

Bits	Name	Memory Access	Description
13:0	AsyncAcRxMode	R/W	<p>AsyncAcRxMode: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.35 AsyncAcTxEn

- **Description:** AsyncAcTxEn: Reserved for PHY training firmware use.
- **Size:** 14 bits
- **Offset:** (0x30000+(j<<12))+0xa2
- **Exists:** Always



**Table 13-300 Fields for Register: AsyncAcTxEn**

Bits	Name	Memory Access	Description
13:0	AsyncAcTxEn	R/W	<p>AsyncAcTxEn: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x3f00</p> <p><b>Exists:</b> Always</p>

### 13.3.36 AsyncAcTxData

- **Description:** AsyncAcTxData: Reserved for PHY training firmware use.
- **Size:** 14 bits
- **Offset:** (0x30000+(j<<12))+0xa3
- **Exists:** Always



**Table 13-301 Fields for Register: AsyncAcTxData**

Bits	Name	Memory Access	Description
13:0	AsyncAcTxData	R/W	<p>AsyncAcTxData: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x2800</p> <p><b>Exists:</b> Always</p>

### 13.3.37 AsyncAcRxData

- **Description:** AsyncAcRxData: Reserved for PHY training firmware use.
- **Size:** 14 bits
- **Offset:** (0x30000+(j<<12))+0xa5
- **Exists:** Always



Table 13-302 Fields for Register: AsyncAcRxData

Bits	Name	Memory Access	Description
13:0	AsyncAcRxData	R	<p>AsyncAcRxData: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.3.38 ForceClkDisable

- **Description:** ForceClkDisable: Clock gating control
- **Size:** 1 bit
- **Offset:** (0x30000+(j<<12))+0xa6
- **Exists:** Always



**Table 13-303 Fields for Register: ForceClkDisable**

Bits	Name	Memory Access	Description
0	ForceClkDisable	R/W	<p>ForceClkDisable: This CSR forces the gating of MEMCLKs driven from the PHY for both the channels. ForceClkDisable - controls BP_DFI[0/1]_CLK_[H/L]. Polarity of clock stop will be defined by csr CkDisVal.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.39 CaBusTriEn

- **Description:** CaBusTriEn: Enable Tri-Stating of the CA bus
- **Size:** 1 bit
- **Offset:** (0x30000+(j<<12))+0xab
- **Exists:** Always



**Table 13-304 Fields for Register: CaBusTriEn**

Bits	Name	Memory Access	Description
0	CaBusTriEn	R/W	<p>CaBusTriEn: Power saving mode for sideband interaction. Tri-Stating of the CA bus CA[5:0] in LPDDR4X and CA[6:0] in LPDDR5.</p> <ul style="list-style-type: none"> <li>■ 0 - CA Bus is not tristate during side band transaction</li> <li>■ 1 - CA Bus is tristate during side band transaction</li> </ul> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

### 13.3.40 AcLnDisable

- **Description:** AcLnDisable: Per AC Lane Control to disable the lane
- **Size:** 13 bits
- **Offset:** (0x30000+(j<<12))+0xac
- **Exists:** Always



**Table 13-305 Fields for Register: AcLnDisable**

Bits	Name	Memory Access	Description
12:0	AcLnDisable	R/W	<p>AcLnDisable: When a bit is set, the corresponding lane is disabled Lanes: LP5 Lanes: Lower Upper</p> <ul style="list-style-type: none"> <li>■ 0 Tx SE lane0 BP[0] BP[9]</li> <li>■ 1 Tx SE lane1 BP[1] BP[10]</li> <li>■ 2 Tx SE lane2 BP[2] BP[11]</li> <li>■ 3 Tx SE lane3 BP[3] BP[12]</li> <li>■ 4 Tx SE lane4 BP[6] BP[15]</li> <li>■ 5 Tx SE lane5 BP[7] BP[16]</li> <li>■ 6 Tx SE lane6 BP[8] BP[17]</li> <li>■ 7 Tx SE lane7 BP.DTO BP[17]</li> <li>■ 8 Tx SEC lane0 BP[4] BP[13]</li> <li>■ 9 Tx SEC lane1 BP[5] BP[14]</li> <li>■ 10 Tx DIFF0 lane</li> <li>■ 11 Unused</li> <li>■ 12 DTO</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.41 DfiClkAcLnDis

- **Description:** DfiClkAcLnDis: Per AC Lane Control to disable the DfiClk in the lane
- **Size:** 13 bits
- **Offset:** (0x30000+(j<<12))+0xad
- **Exists:** Always



**Table 13-306 Fields for Register: DfiClkAcLnDis**

Bits	Name	Memory Access	Description
12:0	DfiClkAcLnDis	R/W	<p>DfiClkAcLnDis: When a bit is set, the corresponding lane's DfiClk is disabled. This field should only be changed PHY initialization step C. Lanes: LP5 Lanes: Lower Upper</p> <ul style="list-style-type: none"> <li>■ 0 Tx SE lane0 BP[0] BP[9]</li> <li>■ 1 Tx SE lane1 BP[1] BP[10]</li> <li>■ 2 Tx SE lane2 BP[2] BP[11]</li> <li>■ 3 Tx SE lane3 BP[3] BP[12]</li> <li>■ 4 Tx SE lane4 BP[6] BP[15]</li> <li>■ 5 Tx SE lane5 BP[7] BP[16]</li> <li>■ 6 Tx SE lane6 BP[8] BP[17]</li> <li>■ 7 Tx SE lane7 BP_DTO BP[17]</li> <li>■ 8 Tx SEC lane0 BP[4] BP[13]</li> <li>■ 9 Tx SEC lane1 BP[5] BP[14]</li> <li>■ 10 Tx DIFF0 lane</li> <li>■ 11 Unused</li> <li>■ 12 DTO</li> </ul> <p><b>Value After Reset:</b> 0x0 <b>Exists:</b> Always</p>

### 13.3.42 PClkAcLnDis

- **Description:** PClkAcLnDis: Per AC Lane Control to disable the PClk in the lane
- **Size:** 13 bits
- **Offset:** (0x30000+(j<<12))+0xae
- **Exists:** Always



**Table 13-307 Fields for Register: PClkAcLnDis**

Bits	Name	Memory Access	Description
12:0	PClkAcLnDis	R/W	<p>PClkAcLnDis: When a bit is set, the corresponding lane's PClk is disabled. This field should only be changed PHY initialization step C. Lanes: LP5 Lanes: Lower Upper</p> <ul style="list-style-type: none"> <li>■ 0 Tx SE lane0 BP[0] BP[9]</li> <li>■ 1 Tx SE lane1 BP[1] BP[10]</li> <li>■ 2 Tx SE lane2 BP[2] BP[11]</li> <li>■ 3 Tx SE lane3 BP[3] BP[12]</li> <li>■ 4 Tx SE lane4 BP[6] BP[15]</li> <li>■ 5 Tx SE lane5 BP[7] BP[16]</li> <li>■ 6 Tx SE lane6 BP[8] BP[17]</li> <li>■ 7 Tx SE lane7 BP.DTO BP[17]</li> <li>■ 8 Tx SEC lane0 BP[4] BP[13]</li> <li>■ 9 Tx SEC lane1 BP[5] BP[14]</li> <li>■ 10 Tx DIFF0 lane</li> <li>■ 11 Unused</li> <li>■ 12 DTO</li> </ul> <p>Corresponding DfiClkAcLnDis bit needs to be 0 when toggling bit in this CSR.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.43 AcLcdlCalPhDetOut

- **Description:** AcLcdlCalPhDetOut: Reserved for PHY training firmware use.
- **Size:** 12 bits
- **Offset:** (0x30000+(j<<12))+0xaf
- **Exists:** Always



**Table 13-308 Fields for Register: AcLcdlCalPhDetOut**

Bits	Name	Memory Access	Description
11:0	AcLcdlCalPhDetOut	R	AcLcdlCalPhDetOut: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always <b>Reset Mask:</b> 0x0 <b>Volatile:</b> true

### 13.3.44 ACXTxDly\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 9)

- **Description:** ACXTxDly\_rY\_pX: Address/Command Delay, per pstate.
- **Size:** 9 bits
- **Offset:** (0x30000+(j<<12))+0xd8+(X\*0x100000)+(Y\*0x100)
- **Exists:** Always



**Table 13-309 Fields for Register: ACXTxDly\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 9)**

Bits	Name	Memory Access	Description
8:0	ACXTxDly_r0_p0	R/W	<p>ACXTxDly_r0_p0: Trained to generate timed address and command signals to the DRAMs, per AC channel</p> <ul style="list-style-type: none"> <li>■ ACXTxDly[8:6] is the coarse delay, i.e., one unit of delay is 1 UI. Range 0 to 3UI, for 1:4 mode a range of 0 to 7UI is supported.</li> <li>■ ACXTxDly[5:0] is the fine delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> </ul> <p>In LPDDR5 mode :</p> <ul style="list-style-type: none"> <li>■ ACXTxDly_r0 controls CA0</li> <li>■ ACXTxDly_r1 controls CA1</li> <li>⋮</li> <li>■ ACXTxDly_r6 controls CA6</li> <li>■ ACXTxDly_r8 controls CS0</li> <li>■ ACXTxDly_r9 controls CS1</li> </ul> <p>In LPDDR4X mode :</p> <ul style="list-style-type: none"> <li>■ ACXTxDly_r0 controls CA0</li> <li>■ ACXTxDly_r1 controls CA1</li> <li>⋮</li> <li>■ ACXTxDly_r5 controls CA5</li> <li>■ ACXTxDly_r6 controls CS0</li> <li>■ ACXTxDly_r7 controls CS1</li> <li>■ ACXTxDly_r8 controls CKE0</li> <li>■ ACXTxDly_r9 controls CKE1</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.45 CKXTxDly\_pX (for X = 0; X <= 3)

- **Description:** CKXTxDly\_pX: Reserved for PHY training firmware use.
- **Size:** 9 bits
- **Offset:** (0x30000+(j<<12))+0xd9+(X\*0x100000)
- **Exists:** Always



**Table 13-310 Fields for Register: CKXTxDly\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
8:0	CKXTxDly_p0	R/W	<p>CKXTxDly_p0: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.46 ACXTxDlyDTO\_pX (for X = 0; X <= 3)

- **Description:** ACXTxDlyDTO\_pX: Address/Command Delay, per pstate.
- **Size:** 9 bits
- **Offset:** (0x30000+(j<<12))+0xda+(X\*0x100000)
- **Exists:** Always



**Table 13-311 Fields for Register: ACXTxDlyDTO\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
8:0	ACXTxDlyDTO_p0	R/W	<p>ACXTxDlyDTO_p0: Trained to generate timed address and command signals to the DRAMs, specifically for the DTO channel</p> <ul style="list-style-type: none"> <li>■ ACXTxDlyDTO[8:6] is the coarse delay, i.e., one unit of delay is 1 UI. Range 0 to 3UI</li> <li>■ ACXTxDlyDTO[5:0] is the fine delay, i.e., one unit of delay is one-sixtyfourth of a UI = UI/64.</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.3.47 ACXTxDly2nMode\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 9)

- **Description:** ACXTxDly2nMode\_rY\_pX: Unused for LP5 mode.
- **Size:** 9 bits
- **Offset:** (0x30000+(j<<12))+0xde+(X\*0x100000)+(Y\*0x100)
- **Exists:** Always



**Table 13-312 Fields for Register: ACXTxDly2nMode\_rY\_pX (for X = 0; X <= 3)(for Y = 0; Y <= 9)**

Bits	Name	Memory Access	Description
8:0	ACXTxDly2nMode_r0_p0	R/W	ACXTxDly2nMode_r0_p0: Not connected when running LP5 mode. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.3.48 AcLcdlUpdInterval\_pX (for X = 0; X <= 3)

- **Description:** AcLcdlUpdInterval\_pX: Controls for the clock ACX2 DLLs
- **Size:** 16 bits
- **Offset:** (0x30000+(j<<12))+0xeb+(X\*0x100000)
- **Exists:** Always



**Table 13-313 Fields for Register: AcLcdlUpdInterval\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	AcLcdlUpdInterval_p0	R/W	<p>AcLcdlUpdInterval_p0: Controls for the clock ACX2 DLLs This contains the interval in DfiClks between updates applied to the TxDLL in the Clock. ACX2s. It affects only DLLs in ACX2s where MDLL is selected. A zero disables updates. Minimum value = 64.</p> <p><b>Value After Reset:</b> 0x80</p> <p><b>Exists:</b> Always</p>

### 13.3.49 LcdlCalSeqUpdCK\_pX (for X = 0; X <= 3)

- **Description:** LcdlCalSeqUpdCK\_pX: Controls the LCDLCALSEQ\_CK LongBubble
- **Size:** 2 bits
- **Offset:** (0x30000+(j<<12))+0xec+(X\*0x100000)
- **Exists:** Always



**Table 13-314 Fields for Register: LcdlCalSeqUpdCK\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
1:0	LcdlCalSeqUpdCK_p0	R/W	<p>LcdlCalSeqUpdCK_p0: Controls the LCDLCALSEQ_CK LongBubble event for CK calibration.</p> <ul style="list-style-type: none"> <li>■ 0: Controlled Longbubble event in the LCDLCALSEQ_CK. Set min 2 DfiClk to start tracking.</li> <li>■ 1: Set 0/1 to disable/enable the PUB Sideband LongBubble event to the LCDLCALSEQ_CK</li> </ul> <p><b>Value After Reset:</b> 0x0 <b>Exists:</b> Always</p>

### 13.3.50 CkTriEn

- **Description:** CkTriEn: Enables the AC CK to Tristate during dram\_clk\_disable
- **Size:** 1 bit
- **Offset:** (0x30000+(j<<12))+0xed
- **Exists:** Always



**Table 13-315 Fields for Register: CkTriEn**

Bits	Name	Memory Access	Description
0	CkTriEn	R/W	<p>CkTriEn: When enabled, the CK_t/c will hit a tri-state value instead of 0/1 when dram_clk_disable.</p> <ul style="list-style-type: none"> <li>■ CkTriEn = 0 : CK_t/c will enter a 0/1 parked state when dram_clk_disable=1</li> <li>■ CkTriEn = 1 : CK_t/c will enter a tri-state value when dram_clk_disable=1</li> </ul> <p><b>Value After Reset:</b> 0x1  <b>Exists:</b> Always</p>

### 13.3.51 ACReservedX (for X == 0)

- **Description:** ACReservedX: Reserved for future use
- **Size:** 16 bits
- **Offset:**  $(0x30000 + (j << 12)) + 0xf9$
- **Exists:** Always



**Table 13-316 Fields for Register: ACReservedX (for X == 0)**

Bits	Name	Memory Access	Description
15:0	ACReserved0	R/W	ACReserved0: Reserved for future use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.3.52 LcdlCalCtrl

- **Description:** LcdlCalCtrl: Reserved for PHY test firmware use.
- **Size:** 6 bits
- **Offset:** (0x30000+(j<<12))+0xfe
- **Exists:** Always



**Table 13-317 Fields for Register: LcdlCalCtrl**

Bits	Name	Memory Access	Description
5:0	LcdlCalCtrl	R/W	LcdlCalCtrl: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.3.53 PUBReservedP1\_pX (AC) (for X = 0; X <= 3)

- **Description:** PUBReservedP1\_pX: Reserved for future use
- **Size:** 8 bits
- **Offset:** (0x30000+(j<<12))+0xff+(X\*0x100000)
- **Exists:** Always



**Table 13-318 Fields for Register: PUBReservedP1\_pX (AC) (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
7:0	PUBReservedP1_p0	R/W	<p>PUBReservedP1_p0: Reserved for future use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.54 PclkDCDCtrl\_pX (AC) (for X = 0; X <= 3)

- **Description:** PclkDCDCtrl\_pX: Controls the DCD Comparator for DCA calibration
- **Size:** 2 bits
- **Offset:** (0x30000+(j<<12))+0x100+(X\*0x100000)
- **Exists:** Always



**Table 13-319 Fields for Register: PclkDCDCtrl\_pX (AC) (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
1	PclkDCDOffsetMode	R/W	PclkDCDOffsetMode: Puts the comparator in offset calibration mode <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
0	PclkDCDEn	R/W	PclkDCDEn: Enables the comparator <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.3.55 ForceInternalUpdate

- **Description:** ForceInternalUpdate: This Register used by Training Firmware to force an internal PHY Update Event.
- **Size:** 1 bit
- **Offset:** (0x30000+(j<<12))+0x103
- **Exists:** Always



**Table 13-320 Fields for Register: ForceInternalUpdate**

Bits	Name	Memory Access	Description
0	ForceInternalUpdate	R/W	<p>ForceInternalUpdate: This Register is used by Training Firmware to force an internal PHY Update Event. Optionally can be used by System Software to issue a PHY update Event. The register may be written to 1'b1 only when all of the following are true:</p> <ul style="list-style-type: none"> <li>■ The DFI interface is offline.</li> <li>■ The Training Hardware is sending only DES and all transactions have retired.</li> </ul> <p>Additionally, this Register must stay asserted for at least 32 DFICLKs before clearing. Prematurely clearing this register may result in an incomplete update.</p> <p><b>Value After Reset:</b> 0x0 <b>Exists:</b> Always</p>

### 13.3.56 ATestPrbsErrCntSECX (for X = 0; X <= 1)

- **Description:** ATestPrbsErrCntSECX: Address Loopback Test Result register
- **Size:** 16 bits
- **Offset:** (0x30000+(j<<12))+0x1a8+(X\*0x1)
- **Exists:** Always



**Table 13-321 Fields for Register: ATestPrbsErrCntSECX (for X = 0; X <= 1)**

Bits	Name	Memory Access	Description
15:0	ATestPrbsErrCntSEC0	R	<p>ATestPrbsErrCntSEC0: Overall error indicator in the prbs checker per se slice. Non-zero value indicating the number of errors seen by self-seeding prbs checker.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.3.57 AcPDsampleSECX (for X = 0; X <= 1)

- **Description:** AcPDsampleSECX: Address Loopback sample values for SEC slice
- **Size:** 1 bit
- **Offset:** (0x30000+(j<<12))+0x1b8+(X\*0x1)
- **Exists:** Always



**Table 13-322 Fields for Register: AcPDsampleSECX (for X = 0; X <= 1)**

Bits	Name	Memory Access	Description
0	AcPDsampleSEC0	R	<p>AcPDsampleSEC0: This returns the sec slice's loopback value sampled by PDClk.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.3.58 AcPDsampleDIFF

- **Name:** Address Loopback sample values for Diff slice
- **Description:** AcPDsampleDIFF: This returns the diff slice's loopback value sampled by PDClk.
- **Size:** 2 bits
- **Offset:**  $(0x30000 + (j \ll 12)) + 0x1ca$
- **Exists:** Always



**Table 13-323 Fields for Register: AcPDsampleDIFF**

Bits	Name	Memory Access	Description
1	PDsampleC	R	PDsampleC: PD0 sample value returned by Complementary bit of Diff slice <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always <b>Reset Mask:</b> 0x0 <b>Volatile:</b> true
0	PDsampleT	R	PDsampleT: PD0 sample value returned by True bit of Diff slice <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always <b>Reset Mask:</b> 0x0 <b>Volatile:</b> true

### 13.3.59 ATestPrbsErrCntDIFF0T

- **Description:** ATestPrbsErrCntDIFF0T: Address Loopback Test Result register
- **Size:** 16 bits
- **Offset:** (0x30000+(j<<12))+0x3e0
- **Exists:** Always



**Table 13-324 Fields for Register: ATestPrbsErrCntDIFF0T**

Bits	Name	Memory Access	Description
15:0	ATestPrbsErrCntDIFF0T	R	<p>ATestPrbsErrCntDIFF0T: Overall error indicator in the prbs checker per se slice. Non-zero value indicating the number of errors seen by self-seeding prbs checker.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.3.60 ATestPrbsErrCntDIFF0C

- **Description:** ATestPrbsErrCntDIFF0C: Address Loopback Test Result register
- **Size:** 16 bits
- **Offset:** (0x30000+(j<<12))+0x3e4
- **Exists:** Always

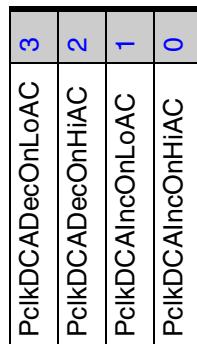


**Table 13-325 Fields for Register: ATestPrbsErrCntDIFF0C**

Bits	Name	Memory Access	Description
15:0	ATestPrbsErrCntDIFF0C	R	<p>ATestPrbsErrCntDIFF0C: Overall error indicator in the prbs checker per se slice. Non-zero value indicating the number of errors seen by self-seeding prbs checker.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.3.61 PclkDCACalCtrl0AC

- **Name:** Controls DCA sample polarity
- **Description:** PclkDCACalCtrl0AC: Controls DCA sample polarity. Used by PHYINIT.
- **Size:** 4 bits
- **Offset:** (0x30000+(j<<12))+0x800
- **Exists:** Always



**Table 13-326 Fields for Register: PclkDCACalCtrl0AC**

Bits	Name	Memory Access	Description
3	PclkDCADecOnLoAC	R/W	PclkDCADecOnLoAC: Decrement DCA value on low samples <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always
2	PclkDCADecOnHiAC	R/W	PclkDCADecOnHiAC: Decrement DCA value on high samples <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
1	PclkDCAIncOnLoAC	R/W	PclkDCAIncOnLoAC: Increment DCA value on low samples <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
0	PclkDCAIncOnHiAC	R/W	PclkDCAIncOnHiAC: Increment DCA value on high samples <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always

### 13.3.62 PclkDCADynCtrl

- **Name:** Dynamic bits for DCA control
- **Description:** PclkDCADynCtrl: Dynamic bits for DCA control. Used by PHYINIT.
- **Size:** 4 bits
- **Offset:** (0x30000+(j<<12))+0x802
- **Exists:** Always

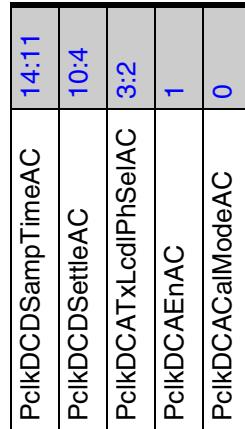


**Table 13-327 Fields for Register: PclkDCADynCtrl**

Bits	Name	Memory Access	Description
3	PclkDCAForceUpd	R/W	PclkDCAForceUpd: Controls code update pin of PCLK RX DCA macro when PclkCalReset=1 <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
2	PclkDCAForceSampVld	R/W	PclkDCAForceSampVld: Forces CalSeq to take a manual sample <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
1	PclkDCAQuickSearch	R/W	PclkDCAQuickSearch: 1 = Do quick search. 0 = Do full search. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
0	PclkDCACalReset	R/W	PclkDCACalReset: De-assert this to begin DCA calibration Must be kept de-asserted until PclkDcaDone=1 <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always

### 13.3.63 PclkDCAStaticCtrl0AC\_pX (for X = 0; X <= 3)

- **Name:** Static bits for DCA control in AC and CK blocks
- **Description:** PclkDCAStaticCtrl0AC\_pX: Static bits for DCA control in AC and CK blocks. Used by PHYINIT.
- **Size:** 15 bits
- **Offset:** (0x30000+(j<<12))+0x803+(X\*0x100000)
- **Exists:** Always



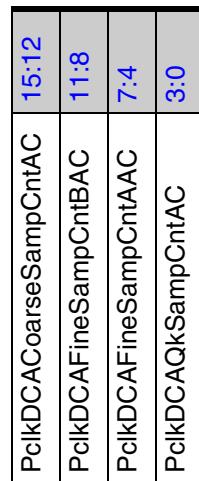
**Table 13-328 Fields for Register: PclkDCAStaticCtrl0AC\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
14:11	PclkDCDSampTimeAC	R/W	<p>PclkDCDSampTimeAC: Set by PHYINIT when PCLK DCA is using Comparator Indicates how long to delay after SampCkEn is issued Delay will be (VALUE) DFICLKs. Reset VALUE is 2.</p> <p><b>Value After Reset:</b> 0x2</p> <p><b>Exists:</b> Always</p>
10:4	PclkDCDSettleAC	R/W	<p>PclkDCDSettleAC: Set by PHYINIT when PCLK DCA is using Comparator Indicates how long to delay after RXDCA code update pulse is issued Delay will [(2*VALUE)-1] DFICLKs. Reset VALUE is 4.</p> <p><b>Value After Reset:</b> 0x4</p> <p><b>Exists:</b> Always</p>
3:2	PclkDCATxLcdlPhSelAC	R/W	<p>PclkDCATxLcdlPhSelAC: Selects the DlySteps input (aka Phase) to lcdl_tx 00 = Use TxClk_Phase_Cal 01 = Use (TxClk_Phase_Cal/2) 10 = Use csrPclkDCATxLcdlPhase 11 = Use value chosen by TxTimingSel</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

<b>Bits</b>	<b>Name</b>	<b>Memory Access</b>	<b>Description</b>
1	PclkDCAEnAC	R/W	PclkDCAEnAC: 1 = Enable DCA if PClkEn=1. 0 = Disable DCA <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
0	PclkDCACalModeAC	R/W	PclkDCACalModeAC: 1 = Use LCDL for PhDet 0 = Use DCD for PhDet <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.3.64 PclkDCASampCntAC

- **Name:** Sample counts for various stages of DCA calibration
- **Description:** PclkDCASampCntAC: Sample counts for various stages of DCA calibration. Used by PHYINIT.
- **Size:** 16 bits
- **Offset:** (0x30000+(j<<12))+0x804
- **Exists:** Always



**Table 13-329 Fields for Register: PclkDCASampCntAC**

Bits	Name	Memory Access	Description
15:12	PclkDCACoarseSampCntAC	R/W	PclkDCACoarseSampCntAC: Number of coarse samples for full search <b>Value After Reset:</b> 0x4 <b>Exists:</b> Always
11:8	PclkDCAFineSampCntBAC	R/W	PclkDCAFineSampCntBAC: Number of fine samples for full search in UR or LL <b>Value After Reset:</b> 0x6 <b>Exists:</b> Always
7:4	PclkDCAFineSampCntAAC	R/W	PclkDCAFineSampCntAAC: Number of fine samples for full search on Grayline <b>Value After Reset:</b> 0x6 <b>Exists:</b> Always
3:0	PclkDCAQkSampCntAC	R/W	PclkDCAQkSampCntAC: Number of samples for quick search <b>Value After Reset:</b> 0x3 <b>Exists:</b> Always

### 13.3.65 PclkDCAHysMaskAC

- **Description:** PclkDCAHysMaskAC: Used by DCA quick search algorithm
- **Size:** 3 bits
- **Offset:** (0x30000+(j<<12))+0x805
- **Exists:** Always



**Table 13-330 Fields for Register: PclkDCAHysMaskAC**

Bits	Name	Memory Access	Description
2:0	PclkDCAHysMaskAC	R/W	PclkDCAHysMaskAC: Set one bit for every quick search sample. Must always be 0x7. <b>Value After Reset:</b> 0x7 <b>Exists:</b> Always

### 13.3.66 PclkDCACalFineBoundAC

- **Name:** Sets limit values of DCA Fine searches
- **Description:** PclkDCACalFineBoundAC: Sets limit values of DCA Fine searches. Used by PHYINIT.
- **Size:** 12 bits
- **Offset:** (0x30000+(j<<12))+0x806
- **Exists:** Always



**Table 13-331 Fields for Register: PclkDCACalFineBoundAC**

Bits	Name	Memory Access	Description
11:9	PclkDCALLMinFineAC	R/W	PclkDCALLMinFineAC: LL Min Fine <b>Value After Reset:</b> 0x6 <b>Exists:</b> Always
8:6	PclkDCALLMaxFineAC	R/W	PclkDCALLMaxFineAC: LL Max Fine <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always
5:3	PclkDCAURMinFineAC	R/W	PclkDCAURMinFineAC: UR Min Fine <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
2:0	PclkDCAURMaxFineAC	R/W	PclkDCAURMaxFineAC: UR Max Fine <b>Value After Reset:</b> 0x6 <b>Exists:</b> Always

### 13.3.67 PclkDCANextFineOnCoarseAC

- **Name:** Sets the DCAFine value to use when Coarse changes
- **Description:** PclkDCANextFineOnCoarseAC: Sets the DCAFine value to use when Coarse changes. Used by PHYINIT.
- **Size:** 16 bits
- **Offset:** (0x30000+(j<<12))+0x807
- **Exists:** Always



**Table 13-332 Fields for Register: PclkDCANextFineOnCoarseAC**

Bits	Name	Memory Access	Description
15:12	PclkDCACoarseDecFineLLAC	R/W	PclkDCACoarseDecFineLLAC: Use this when coarse decrements in LL <b>Value After Reset:</b> 0xc <b>Exists:</b> Always
11:8	PclkDCACoarseIncFineLLAC	R/W	PclkDCACoarseIncFineLLAC: Use this when coarse increments in LL <b>Value After Reset:</b> 0xa <b>Exists:</b> Always
7:4	PclkDCACoarseDecFineURAC	R/W	PclkDCACoarseDecFineURAC: Use this when coarse decrements in UR <b>Value After Reset:</b> 0x2 <b>Exists:</b> Always
3:0	PclkDCACoarseIncFineURAC	R/W	PclkDCACoarseIncFineURAC: Use this when coarse increments in UR <b>Value After Reset:</b> 0x4 <b>Exists:</b> Always

### 13.3.68 PclkDCAFullSearchIVACAC

- **Name:** Initial DCA Fine value to use for full searches
- **Description:** PclkDCAFullSearchIVACAC: Initial DCA Fine value to use for full searches. Used by PHYINIT.
- **Size:** 8 bits
- **Offset:** (0x30000+(j<<12))+0x808
- **Exists:** Always

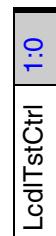


**Table 13-333 Fields for Register: PclkDCAFullSearchIVACAC**

Bits	Name	Memory Access	Description
7:4	PclkDCAFineIVMinAC	R/W	PclkDCAFineIVMinAC: Fine value for Samp 2 <b>Value After Reset:</b> 0xe <b>Exists:</b> Always
3:0	PclkDCAFineIVMaxAC	R/W	PclkDCAFineIVMaxAC: Fine value for Samp 1 <b>Value After Reset:</b> 0x6 <b>Exists:</b> Always

### 13.3.69 LcdITstCtrl

- **Description:** LcdITstCtrl: Reserved for PHY test firmware use.
- **Size:** 2 bits
- **Offset:** (0x30000+(j<<12))+0x884
- **Exists:** Always



**Table 13-334 Fields for Register: LcdITstCtrl**

Bits	Name	Memory Access	Description
1:0	LcdITstCtrl	R/W	LcdITstCtrl: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.3.70 AcLoopBackEnLnX (for X = 0; X <= 10)

- **Description:** AcLoopBackEnLnX: Per AC Lane, Loopback Enable Configuration
- **Size:** 1 bit
- **Offset:** (0x30000+(j<<12))+0x900+(X\*0x1)
- **Exists:** Always



**Table 13-335 Fields for Register: AcLoopBackEnLnX (for X = 0; X <= 10)**

Bits	Name	Memory Access	Description
0	AcLoopBackEnLn0	R/W	<p>AcLoopBackEnLn0: This field is programmed as follows:</p> <ul style="list-style-type: none"> <li>■ 0: Mission Mode</li> <li>■ 1: HWT LoopBackMode, Set csrAcCoreLoopBackMode accordingly for PadSide or CoreSide LoopBack. LP5 Lanes: Lower Upper</li> <li>■ Ln 0 Tx SE lane0 BP[0] BP[10]</li> <li>■ Ln 1 Tx SE lane1 BP[1] BP[11]</li> <li>■ Ln 2 Tx SE lane2 BP[2] BP[12]</li> <li>■ Ln 3 Tx SE lane3 BP[3] BP[13]</li> <li>■ Ln 4 Tx SE lane4 BP[6] BP[16]</li> <li>■ Ln 5 Tx SE lane5 BP[7] BP[17]</li> <li>■ Ln 6 Tx SE lane6 BP[8] BP[18]</li> <li>■ Ln 7 Tx SE lane7 BP[9] BP[19]</li> <li>■ Ln 8 Tx SEC lane0 BP[4] BP[14]</li> <li>■ Ln 9 Tx SEC lane1 BP[5] BP[15]</li> <li>■ Ln 10 CK0</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.3.71 ATestPrbsErrCntSEX (for X = 0; X <= 7)

- **Description:** ATestPrbsErrCntSEX: Address Loopback Test Result register
- **Size:** 16 bits
- **Offset:** (0x30000+(j<<12))+0xb00+(X\*0x1)
- **Exists:** Always



**Table 13-336 Fields for Register: ATestPrbsErrCntSEX (for X = 0; X <= 7)**

Bits	Name	Memory Access	Description
15:0	ATestPrbsErrCntSE0	R	<p>ATestPrbsErrCntSE0: Overall error indicator in the prbs checker per se slice. Non-zero value indicating the number of errors seen by self-seeding prbs checker.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

## 13.4 **DWC\_DDRPHYA\_PPGCj\_Pk Registers**

### 13.4.1 PpgcGenCtrl

- **Description:** PpgcGenCtrl: Reserved for PHY training firmware use.
- **Size:** 1 bit
- **Offset:** 0x70000+0x0
- **Exists:** Always



**Table 13-337 Fields for Register: PpgcGenCtrl**

Bits	Name	Memory Access	Description
0	PpgcGenCtrl	R/W	PpgcGenCtrl: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.2 PpgcGenDbiCtrl

- **Description:** PpgcGenDbiCtrl: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** 0x70000+0x1
- **Exists:** Always



**Table 13-338 Fields for Register: PpgcGenDbiCtrl**

Bits	Name	Memory Access	Description
15:0	PpgcGenDbiCtrl	R/W	PpgcGenDbiCtrl: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.3 PpgcGenDbiConfig

- **Description:** PpgcGenDbiConfig: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** 0x70000+0x2
- **Exists:** Always



**Table 13-339 Fields for Register: PpgcGenDbiConfig**

Bits	Name	Memory Access	Description
15:0	PpgcGenDbiConfig	R/W	PpgcGenDbiConfig: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.4 PpgcGenLaneMuxSelX (for X = 0; X <= 1)

- **Description:** PpgcGenLaneMuxSelX: Reserved for PHY training firmware use.
- **Size:** 9 bits
- **Offset:** 0x70000+0x3+(X\*0x1)
- **Exists:** Always



**Table 13-340 Fields for Register: PpgcGenLaneMuxSelX (for X = 0; X <= 1)**

Bits	Name	Memory Access	Description
8:0	PpgcGenLaneMuxSel0	R/W	PpgcGenLaneMuxSel0: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.5 EnPhyUpdZQCalUpdate

- **Description:** EnPhyUpdZQCalUpdate: Enable ZQ Cal on Phy Update
- **Size:** 2 bits
- **Offset:** 0x70000+0x5
- **Exists:** Always

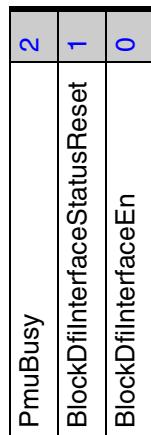


**Table 13-341 Fields for Register: EnPhyUpdZQCalUpdate**

Bits	Name	Memory Access	Description
1:0	EnPhyUpdZQCalUpdate	R/W	<p>EnPhyUpdZQCalUpdate: When set, Enables (per-channel) Zqupdate during ctrlupd and phyupd events. Bit 0 for channel 0, Bit 1 for channel 1.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.6 BlockDfiInterface

- **Description:** BlockDfiInterface: Used to block certain operations when PState info is being transferred from DCCM
- **Size:** 3 bits
- **Offset:** 0x70000+0x6
- **Exists:** Always



**Table 13-342 Fields for Register: BlockDfiInterface**

Bits	Name	Memory Access	Description
2	PmuBusy	R/W	<p>PmuBusy: Directly controls the _pmu_busy top-level pin</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
1	BlockDfiInterfaceStatusReset	R/W	<p>BlockDfiInterfaceStatusReset: When this bit is set, it clears the sticky bits in csrBlockDfiInterfaceStatus.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
0	BlockDfiInterfaceEn	R/W	<p>BlockDfiInterfaceEn: When set, blocks operations normally triggered by dfi_init_start, dfi_ctrlupd_req, dfi_lp_ctrl_req, and dfi_lp_data_req. Also disables the Phy Master Interface and the PIE. When csrBlockDfiInterface=1, PHY internally blocks starting of PIE for all cases:</p> <ul style="list-style-type: none"> <li>■ DFI frequency change request from UMC or PMI timer based</li> <li>■ PHY doesn't issue dfi_phyupd_req or dfi_phymstr_req</li> <li>■ PHY doesn't acknowledge dfi_ctrlupd_req</li> <li>■ PHY doesn't acknowledge dfi_lp_ctrl_req, dfi_lp_data_req</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.7 BlockDfiInterfaceStatus

- **Description:** BlockDfiInterfaceStatus: Tracks blocked assertions of sideband signals when csrBlockDfiInterfaceEn=1
- **Size:** 8 bits
- **Offset:** 0x70000+0x7
- **Exists:** Always



**Table 13-343 Fields for Register: BlockDfiInterfaceStatus**

Bits	Name	Memory Access	Description
7:0	BlockDfiInterfaceStatus	R	<p>BlockDfiInterfaceStatus: These status bits are sticky and should be cleared by toggling csrBlockDfiInterfaceStatusReset. When BlockDfiInterfaceEn is set, operations normally triggered by certain sideband signals are blocked. This register records whether there was at least one assertion of each sideband signal during this time as follows:</p> <ul style="list-style-type: none"> <li>■ Bit 7: dfi_init_start Ch1</li> <li>■ Bit 6: dfi_ctrlupd_req Ch1</li> <li>■ Bit 5: dfi_lp_ctrl_req Ch1</li> <li>■ Bit 4: dfi_lp_data_req Ch1</li> <li>■ Bit 3: dfi_init_start Ch0</li> <li>■ Bit 2: dfi_ctrlupd_req Ch0</li> <li>■ Bit 1: dfi_lp_ctrl_req Ch0</li> <li>■ Bit 0: dfi_lp_data_req Ch0</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always  <b>Reset Mask:</b> 0x0  <b>Volatile:</b> true</p>

### 13.4.8 DfiCustMode\_pX (for X = 0; X <= 3)

- **Description:** DfiCustMode\_pX: Deprecated
- **Size:** 1 bit
- **Offset:** 0x70000+0xb+(X\*0x100000)
- **Exists:** Always



**Table 13-344 Fields for Register: DfiCustMode\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
0	DfiCustMode_p0	R/W	<p>DfiCustMode_p0: Deprecated</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.9 HwtMRL\_pX (for X = 0; X <= 3)

- **Description:** HwtMRL\_pX: HWT MaxReadLatency.
- **Size:** 6 bits
- **Offset:** 0x70000+0xd+(X\*0x100000)
- **Exists:** Always



**Table 13-345 Fields for Register: HwtMRL\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
5:0	HwtMRL_p0	R/W	<p>HwtMRL_p0: This Max Read Latency CSR is to be trained to ensure the rx-data FIFO is not read until after all dbytes have their read data valid. The CSR is in units of two mem clocks; that is, a unit change in the LSB is a change in MRL of 1 DfiClk. This firmware copy of MRL is used by the PHY training hardware only. Maximum supported value is 39 decimal.</p> <p><b>Value After Reset:</b> 0x6</p> <p><b>Exists:</b> Always</p>

### 13.4.10 RegRet

- **Description:** RegRet: Reserved for PHY training firmware use.
- **Size:** 1 bit
- **Offset:** 0x70000+0xe
- **Exists:** Always



**Table 13-346 Fields for Register: RegRet**

Bits	Name	Memory Access	Description
0	RegRet	R/W	<p>RegRet: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.11 DisableZQupdateOnSnoop

- **Description:** DisableZQupdateOnSnoop: Disable ZQUpdate (part of ctrlupd) if snoop\_osc\_running asserted
- **Size:** 1 bit
- **Offset:** 0x70000+0xf
- **Exists:** Always



**Table 13-347 Fields for Register: DisableZQupdateOnSnoop**

Bits	Name	Memory Access	Description
0	DisableZQupdateOnSnoop	R/W	<p>DisableZQupdateOnSnoop: When bit is set, snoop_osc_running will disable the ZQUpdate and StopCK for the -next- ctrlupd event</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.12 Prbs0GenModeSel

- **Description:** Prbs0GenModeSel: Reserved for PHY training firmware use.
- **Size:** 2 bits
- **Offset:** 0x70000+0x10
- **Exists:** Always

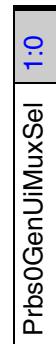


**Table 13-348 Fields for Register: Prbs0GenModeSel**

Bits	Name	Memory Access	Description
1:0	Prbs0GenModeSel	R/W	Prbs0GenModeSel: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.13 Prbs0GenUiMuxSel

- **Description:** Prbs0GenUiMuxSel: Reserved for PHY training firmware use.
- **Size:** 2 bits
- **Offset:** 0x70000+0x11
- **Exists:** Always



**Table 13-349 Fields for Register: Prbs0GenUiMuxSel**

Bits	Name	Memory Access	Description
1:0	Prbs0GenUiMuxSel	R/W	Prbs0GenUiMuxSel: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.14 Prbs0GenTapDlyX (for X = 0; X <= 7)

- **Description:** Prbs0GenTapDlyX: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** 0x70000+0x12+(X\*0x1)
- **Exists:** Always



**Table 13-350 Fields for Register: Prbs0GenTapDlyX (for X = 0; X <= 7)**

Bits	Name	Memory Access	Description
15:0	Prbs0GenTapDly0	R/W	<p>Prbs0GenTapDly0: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.15 MtestMuxSel

- **Description:** MtestMuxSel: Digital Observation Pin control
- **Size:** 10 bits
- **Offset:** 0x70000+0x1a
- **Exists:** Always



**Table 13-351 Fields for Register: MtestMuxSel**

Bits	Name	Memory Access	Description
9:0	MtestMuxSel	R/W	<p>MtestMuxSel: Controls for the mux for asynchronous data to the Digital Observation Pin.</p> <ul style="list-style-type: none"> <li>■ Encoding 9'h0 causes this chiplet to drive 0, (allowing flat 'OR' of pass-through information).</li> <li>■ MtestMuxSel[4:0] - Lower 5 bits selects one bit from the 32 bit MtestMux in each section or slice.</li> </ul> <p>Detailed tables are in the PUB Databook for PUB sections (AC, MASTER, etc.)</p> <ul style="list-style-type: none"> <li>■ MtestMuxSel[8:5] - Where more than one MtestMux exists, non-zero values select the outputs of the additional Mtest-Muxes,           <ul style="list-style-type: none"> <li>□ DBYTE MtestMuxSel[6:5]=2'h0 for Mux-A and 2'h1 for Mux-B and 2'h2 for Mux-C.</li> <li>□ For all other slaves, MtestMuxSel[8:5] are unused</li> </ul> </li> </ul> <p><b>Note:</b> See the PUB Databook for how, or if, the Digital Observation Pin is mapped to a physical bump in this configuration.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.16 Prbs0GenStateLo

- **Description:** Prbs0GenStateLo: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** 0x70000+0x1b
- **Exists:** Always



**Table 13-352 Fields for Register: Prbs0GenStateLo**

Bits	Name	Memory Access	Description
15:0	Prbs0GenStateLo	R/W	Prbs0GenStateLo: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.17 Prbs0GenStateHi

- **Description:** Prbs0GenStateHi: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** 0x70000+0x1c
- **Exists:** Always



**Table 13-353 Fields for Register: Prbs0GenStateHi**

Bits	Name	Memory Access	Description
15:0	Prbs0GenStateHi	R/W	Prbs0GenStateHi: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.18 Prbs1GenModeSel

- **Description:** Prbs1GenModeSel: Reserved for PHY training firmware use.
- **Size:** 2 bits
- **Offset:** 0x70000+0x20
- **Exists:** Always



**Table 13-354 Fields for Register: Prbs1GenModeSel**

Bits	Name	Memory Access	Description
1:0	Prbs1GenModeSel	R/W	<p>Prbs1GenModeSel: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.19 Prbs1GenUiMuxSel

- **Description:** Prbs1GenUiMuxSel: Reserved for PHY training firmware use.
- **Size:** 2 bits
- **Offset:** 0x70000+0x21
- **Exists:** Always



**Table 13-355 Fields for Register: Prbs1GenUiMuxSel**

Bits	Name	Memory Access	Description
1:0	Prbs1GenUiMuxSel	R/W	Prbs1GenUiMuxSel: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.20 Prbs1GenTapDlyX (for X = 0; X <= 7)

- **Description:** Prbs1GenTapDlyX: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** 0x70000+0x22+(X\*0x1)
- **Exists:** Always



**Table 13-356 Fields for Register: Prbs1GenTapDlyX (for X = 0; X <= 7)**

Bits	Name	Memory Access	Description
15:0	Prbs1GenTapDly0	R/W	<p>Prbs1GenTapDly0: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.21 Prbs1GenStateLo

- **Description:** Prbs1GenStateLo: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** 0x70000+0x2b
- **Exists:** Always



**Table 13-357 Fields for Register: Prbs1GenStateLo**

Bits	Name	Memory Access	Description
15:0	Prbs1GenStateLo	R/W	Prbs1GenStateLo: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.22 Prbs1GenStateHi

- **Description:** Prbs1GenStateHi: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** 0x70000+0x2c
- **Exists:** Always



**Table 13-358 Fields for Register: Prbs1GenStateHi**

Bits	Name	Memory Access	Description
15:0	Prbs1GenStateHi	R/W	<p>Prbs1GenStateHi: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.23 Prbs2GenModeSel

- **Description:** Prbs2GenModeSel: Reserved for PHY training firmware use.
- **Size:** 2 bits
- **Offset:** 0x70000+0x30
- **Exists:** Always



**Table 13-359 Fields for Register: Prbs2GenModeSel**

Bits	Name	Memory Access	Description
1:0	Prbs2GenModeSel	R/W	Prbs2GenModeSel: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.24 Prbs2GenUiMuxSel

- Description:** Prbs2GenUiMuxSel: Reserved for PHY training firmware use.
- Size:** 2 bits
- Offset:** 0x70000+0x31
- Exists:** Always



**Table 13-360 Fields for Register: Prbs2GenUiMuxSel**

Bits	Name	Memory Access	Description
1:0	Prbs2GenUiMuxSel	R/W	<p>Prbs2GenUiMuxSel: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.25 Prbs2GenTapDlyX (for X = 0; X <= 7)

- **Description:** Prbs2GenTapDlyX: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** 0x70000+0x32+(X\*0x1)
- **Exists:** Always



**Table 13-361 Fields for Register: Prbs2GenTapDlyX (for X = 0; X <= 7)**

Bits	Name	Memory Access	Description
15:0	Prbs2GenTapDly0	R/W	Prbs2GenTapDly0: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.26 Prbs2GenStateLo

- **Description:** Prbs2GenStateLo: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** 0x70000+0x3b
- **Exists:** Always



**Table 13-362 Fields for Register: Prbs2GenStateLo**

Bits	Name	Memory Access	Description
15:0	Prbs2GenStateLo	R/W	<p>Prbs2GenStateLo: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.27 Prbs2GenStateHi

- **Description:** Prbs2GenStateHi: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** 0x70000+0x3c
- **Exists:** Always



**Table 13-363 Fields for Register: Prbs2GenStateHi**

Bits	Name	Memory Access	Description
15:0	Prbs2GenStateHi	R/W	Prbs2GenStateHi: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.28 PPTTrainSetup\_pX (for X = 0; X <= 3)

- **Name:** Setup Intervals for DFI PHY Master operations
- **Description:** PPTTrainSetup\_pX: Controls for PHY MASTER interface. For LPDDR4X/5 some tracking is required periodically using the DFI PHY MASTER interface. This register control the interval. Setting the register to 0 completely disables the PHY Master Interface. Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support LPDDR4X/5 DRAM drift compensation.
- **Size:** 7 bits
- **Offset:** 0x70000+0x40+(X\*0x100000)
- **Exists:** Always



**Table 13-364 Fields for Register: PPTTrainSetup\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
6:4	PhyMstrMaxReqToAck	R/W	<p>PhyMstrMaxReqToAck: Bits 6:4 of this register specify the max time from tdfi_phymstr_req asserted to tdfi_phymstr_ack asserted. If the request is not acknowledged by this time, dfi_error will be asserted.</p> <ul style="list-style-type: none"> <li>■ 3'b000 Disable PHY Master Interface</li> <li>■ 3'b001 set tPHYMSTR_resp. = 256 DfiClks</li> <li>■ 3'b010 set tPHYMSTR_resp. = 512 DfiClks</li> <li>■ 3'b011 set tPHYMSTR_resp. = 1024 DfiClks</li> <li>■ 3'b100 set tPHYMSTR_resp. = 2048 DfiClks</li> <li>■ 3'b101 set tPHYMSTR_resp. = 4096 DfiClks</li> <li>■ 3'b110 set tPHYMSTR_resp. = 16384 DfiClks</li> <li>■ 3'b111 set tPHYMSTR_resp. = undefined</li> </ul> <p><b>Note 1:</b> The above are nominal values. The actual values are calculated as (nominal_value - 6 - DWC_LPDDR5XPHY_PIPE_DFI_MISC)</p> <p><b>Note 2:</b> This signal interacts with dfi_phyupd_req. See the section "DFI Sideband Simultaneous Collisions" in the PUB Databook.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
3:0	PhyMstrTrainInterval	R/W	<p>PhyMstrTrainInterval: Bits 3:0 of this register specifies the time between the end of one training and the start of the next. (As per section 11 of the Preliminary DFI 4.0 Specification, V2, it is the max expected time from dfi_init_complete asserted to tdfi_phymstr_ack asserted).</p> <ul style="list-style-type: none"> <li>■ 4'b0000 Disable PHY Master Interface</li> <li>■ 4'b0001 PHY MASTER Request Interval = 262144 DfiClks</li> <li>■ 4'b0010 PHY MASTER Request Interval = 524288 DfiClks</li> <li>■ 4'b0011 PHY MASTER Request Interval = 1048576 DfiClks</li> <li>■ 4'b0100 PHY MASTER Request Interval = 2097152 DfiClks</li> <li>■ 4'b0101 PHY MASTER Request Interval = 4194304 DfiClks</li> <li>■ 4'b0110 PHY MASTER Request Interval = 8388608 DfiClks</li> <li>■ 4'b0111 PHY MASTER Request Interval = 16777216 DfiClks</li> <li>■ 4'b1000 PHY MASTER Request Interval = 33554432 DfiClks</li> <li>■ 4'b1001 PHY MASTER Request Interval = 67108864 DfiClks</li> <li>■ 4'b1010 PHY MASTER Request Interval = 134217728 DfiClks</li> <li>■ 4'b1011 - 4'b1111 PHY MASTER Request Interval = undefined</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.4.29 PhyMstrFreqOverride\_pX (for X = 0; X <= 3)

- **Description:** PhyMstrFreqOverride\_pX: Programmed by PHY training firmware to support LPDDR4X DRAM drift compensation.
- **Size:** 5 bits
- **Offset:** 0x70000+0x41+(X\*0x100000)
- **Exists:** Always



**Table 13-365 Fields for Register: PhyMstrFreqOverride\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
4:0	PhyMstrFreqOverride_p0	R/W	<p>PhyMstrFreqOverride_p0: Programmed by PHY training firmware to support LPDDR4X DRAM drift compensation.</p> <p><b>Value After Reset:</b> 0x2</p> <p><b>Exists:</b> Always</p>

### 13.4.30 DfInitComplete

- **Description:** DfInitComplete: DFI Init Complete control
- **Size:** 1 bit
- **Offset:** 0x70000+0x49
- **Exists:** Always



**Table 13-366 Fields for Register: DfInitComplete**

Bits	Name	Memory Access	Description
0	DfInitComplete	R/W	<p>DfInitComplete: This register directly controls DfInitComplete, and is meant to be written by the PState sequencer as part of the power state switching sequence. It should be written to a 0 once the transition to LP2 is complete. Likewise it should be written to 1 once the transition to an active and PState is complete. It should reset to a 0 on both warm and cold reset.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.31 PPGCParityInvert

- **Description:** PPGCParityInvert: Invert APB Parity for register slave PPGC
- **Size:** 2 bits
- **Offset:** 0x70000+0x4d
- **Exists:** Always



**Table 13-367 Fields for Register: PPGCParityInvert**

Bits	Name	Memory Access	Description
1:0	PPGCParityInvert	R/W	<p>PPGCParityInvert: Invert APB Parity for register slave PPGC. As required for Automotive. NOTE: This register should be used only for test. Set the bits for only one slave at a time. When bits are set for a particular slave. APB Reads of only that slave are valid. Bit 0 applies to [7:0] Bit 1 applies to [15:8] In HMZCAL, When ZcalStopClk=1, this csr can not be accessed.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.32 PMIEnable

- **Description:** PMIEnable: This register is dynamically written by PHY Initialization Engine during frequency changes and should not be written by the user.
- **Size:** 1 bit
- **Offset:** 0x70000+0x54
- **Exists:** Always

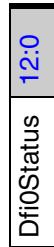


**Table 13-368 Fields for Register: PMIEnable**

Bits	Name	Memory Access	Description
0	PMIEnable	R/W	<p>PMIEnable: This register is dynamically written by PHY Initialization Engine during frequency changes and should not be written by the user.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.33 Dfi0Status

- **Description:** Dfi0Status: Current state of certain Dfi Inputs
- **Size:** 13 bits
- **Offset:** 0x70000+0x5a
- **Exists:** Always

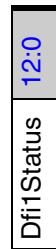


**Table 13-369 Fields for Register: Dfi0Status**

Bits	Name	Memory Access	Description
12:0	Dfi0Status	R	<p>Dfi0Status: Current value of dfi_ inputs as shown:</p> <ul style="list-style-type: none"> <li>■ Bit 0: dfi0_init_start</li> <li>■ Bit 1: dfi0_init_complete</li> <li>■ Bit 3:2: 00</li> <li>■ Bit 5:4: dfi0_freq_ratio</li> <li>■ Bit 7:6: dfi0_freq_fsp</li> <li>■ Bit 12:8: dfi0_frequency</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always  <b>Reset Mask:</b> 0x0  <b>Volatile:</b> true</p>

### 13.4.34 Dfi1Status

- **Description:** Dfi1Status: Current state of certain Dfi Inputs
- **Size:** 13 bits
- **Offset:** 0x70000+0x5b
- **Exists:** Always



**Table 13-370 Fields for Register: Dfi1Status**

Bits	Name	Memory Access	Description
12:0	Dfi1Status	R	<p>Dfi1Status: Current value of dfi_ inputs as shown:</p> <ul style="list-style-type: none"> <li>■ Bit 0: dfi1_init_start</li> <li>■ Bit 1: dfi1_init_complete</li> <li>■ Bit 3:2: 00</li> <li>■ Bit 5:4: dfi1_freq_ratio</li> <li>■ Bit 7:6: dfi1_freq_fsp</li> <li>■ Bit 12:8: dfi1_frequency</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always  <b>Reset Mask:</b> 0x0  <b>Volatile:</b> true</p>

### 13.4.35 DfiHandshakeDelays0\_pX (for X = 0; X <= 3)

- **Name:** Small delays on handshake signals per AC Channel
- **Description:** DfiHandshakeDelays0\_pX: Add assertion/deassertion delays on handshake signals. Logic assumes that dfi signal assertions exceed the programmed delays
- **Size:** 12 bits
- **Offset:** 0x70000+0x66+(X\*0x100000)
- **Exists:** Always



**Table 13-371 Fields for Register: DfiHandshakeDelays0\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
11:8	CtrlUpdReqDelay0	R/W	<p>CtrlUpdReqDelay0: Adds 0-15 DfiClks of additional delay after dfi_ctrlupd_req asserts, before the PHY takes any action (such as starting DDL calibration).</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
7:4	PhyUpdReqDelay0	R/W	<p>PhyUpdReqDelay0: For Debug Use Only. Must be 0. Adds 0-15 DfiClks of additional delay after the PHY completes all PHY update activities, before de-asserting dfi_phyupd_req.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
3:0	PhyUpdAckDelay0	R/W	<p>PhyUpdAckDelay0: For Debug Use Only. Must be 0. Adds 0-15 DfiClks of additional delay after dfi_phyupd_ack asserts, before the PHY takes any action (such as starting DDL calibration).</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.36 DFIPHYUPDX (for X = 0; X <= 1)

- **Name:** Per AC Ch DFIPHYUpd Request time cntr (in DfiClk)
- **Description:** DFIPHYUPDX: This Register is used to determine the DFI Update request interval and the maximum update response time.  
**Note:** DFIPHYUPDCNT and DFIPHYUPDRESP must be chosen such that DFIPHYUPDCNT > DFIPHYUPDRESP
- **Size:** 7 bits
- **Offset:** 0x70000+0x67+(X\*0x80)
- **Exists:** Always



**Table 13-372 Fields for Register: DFIPHYUPDX (for X = 0; X <= 1)**

Bits	Name	Memory Access	Description
6:4	DFIPHYUPDRESP0	R/W	<p>DFIPHYUPDRESP0: Enforces the t_phyupd_resp time, the maximum time that is allowed to controller to respond to the request for a PHY update. A dfi_error will be signaled if there is no acknowledgement of the update request within nDfiClks_phyupd_resp.</p> <ul style="list-style-type: none"> <li>■ 3'b000 nDfiClks_phyupd_resp= 512 DfiClk default value</li> <li>■ 3'b001 nDfiClks_phyupd_resp= 1024 DfiClk</li> <li>■ 3'b010 nDfiClks_phyupd_resp= 2048 DfiClk</li> <li>■ 3'b011 nDfiClks_phyupd_resp= 4096 DfiClk</li> <li>■ 3'b100 nDfiClks_phyupd_resp= 8192 DfiClk</li> </ul> <p><b>Note 1:</b> The above are nominal values. The actual values are reduced by 4.</p> <p><b>Note 2:</b> Any code not enumerated above is RSVD and should not be set.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
3:0	DFIPHYUPDCNT0	R/W	<p>DFIPHYUPDCNT0: This controls the interval between the end of a PHY update transaction and a subsequent request.</p> <ul style="list-style-type: none"> <li>■ 4'b0000 - Disable timer-based Phy Update</li> <li>■ 4'b1001 - 1K DfiClks minus nDfiClks_phyupd_resp</li> <li>■ 4'b1010 - 2K DfiClks minus nDfiClks_phyupd_resp</li> <li>■ 4'b1011 - 4K DfiClks minus nDfiClks_phyupd_resp</li> <li>■ 4'b0001 - 8K DfiClks minus nDfiClks_phyupd_resp</li> <li>■ 4'b0011 - 16K DfiClks minus nDfiClks_phyupd_resp</li> <li>■ 4'b0111 - 32K DfiClks minus nDfiClks_phyupd_resp</li> <li>■ 4'b1111 - 64K DfiClks minus nDfiClks_phyupd_resp</li> </ul> <p>Any code not enumerated above is RSVD and should not be set.</p> <p><b>Value After Reset:</b> 0x7</p> <p><b>Exists:</b> Always</p>

### 13.4.37 DfiLpCtrlEnX (for X = 0; X <= 1)

- **Description:** DfiLpCtrlEnX: DFI LP Ctrl Request power saving controls.
- **Size:** 1 bit
- **Offset:** 0x70000+0x68+(X\*0x80)
- **Exists:** Always



**Table 13-373 Fields for Register: DfiLpCtrlEnX (for X = 0; X <= 1)**

Bits	Name	Memory Access	Description
0	DfiLpCtrlEn0	R/W	<p>DfiLpCtrlEn0: Power saving mode for dfi_lp_ctrl_req</p> <ul style="list-style-type: none"> <li>■ 0 - No Power Saving</li> <li>■ 1 - Power Saving based on the dfi_lp_ctrl_wakeup.</li> </ul> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

### 13.4.38 DfiLpDataEnX (for X = 0; X <= 1)

- **Description:** DfiLpDataEnX: DFI LP Data Request power saving controls.
- **Size:** 1 bit
- **Offset:** 0x70000+0x69+(X\*0x80)
- **Exists:** Always



**Table 13-374 Fields for Register: DfiLpDataEnX (for X = 0; X <= 1)**

Bits	Name	Memory Access	Description
0	DfiLpDataEn0	R/W	<p>DfiLpDataEn0: Power saving mode for dfi_lp_data_req</p> <ul style="list-style-type: none"> <li>■ 0 - No Power Saving.</li> <li>■ 1 - Max Power Saving.</li> </ul> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

### 13.4.39 DynOdtEnCntrlX (for X = 0; X <= 1)

- **Name:** Dynamically Disable the ODT during low power
- **Description:** DynOdtEnCntrlX: Power saving feature
- **Size:** 1 bit
- **Offset:** 0x70000+0x6a+(X\*0x80)
- **Exists:** Always

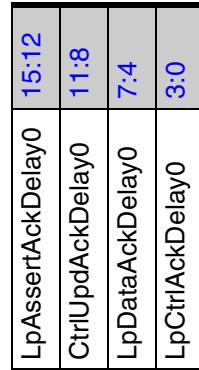


**Table 13-375 Fields for Register: DynOdtEnCntrlX (for X = 0; X <= 1)**

Bits	Name	Memory Access	Description
0	DbyteDynOdtEn0	R/W	<p>DbyteDynOdtEn0: This field is programmed as follows:</p> <ul style="list-style-type: none"> <li>■ When set to 1, OdtEn for all the slices in the corresponding DBYTE channel are dynamically controlled when entering dfi_lp_data_req (DFI Low Power)</li> <li>■ When set to 0, OdtEn for all the slices in the corresponding DBYTE channel are left to its original state.</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.4.40 DfiRespHandshakeDelays0\_pX (for X = 0; X <= 3)

- **Name:** Small delays on handshake signals per AC Channel
- **Description:** DfiRespHandshakeDelays0\_pX: Add assertion/deassertion delays on handshake signals. Logic assumes that dfi signal assertions exceed the programmed delays
- **Size:** 16 bits
- **Offset:** 0x70000+0x6b+(X\*0x100000)
- **Exists:** Always



**Table 13-376 Fields for Register: DfiRespHandshakeDelays0\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:12	LpAssertAckDelay0	R/W	<p>LpAssertAckDelay0: For Debug Use Only. Must be 0. Adds 0-15 DfiClks of additional delay before the PHY would normally assert dfi_lp_data_ack.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
11:8	CtrlUpdAckDelay0	R/W	<p>CtrlUpdAckDelay0: For Debug Use Only. Must be 0. Adds 0-15 DfiClks of additional delay after the PHY completes all PHY update activities, before de-asserting dfi_ctrlupd_ack.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
7:4	LpDataAckDelay0	R/W	<p>LpDataAckDelay0: Adds 0-15 DfiClks of additional delay after the PHY would normally deassert dfi_lp_data_ack.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
3:0	LpCtrlAckDelay0	R/W	<p>LpCtrlAckDelay0: Adds 0-15 DfiClks of additional delay after the PHY would normally deassert dfi_lp_ctrl_ack.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.41 HwtLpCsEnA

- **Description:** HwtLpCsEnA: Programmed by PHY training firmware.
- **Size:** 2 bits
- **Offset:** 0x70000+0x72
- **Exists:** Always



**Table 13-377 Fields for Register: HwtLpCsEnA**

Bits	Name	Memory Access	Description
1:0	HwtLpCsEnA	R/W	<p>HwtLpCsEnA: Programmed by PHY training firmware.</p> <p><b>Value After Reset:</b> 0x3</p> <p><b>Exists:</b> Always</p>

### 13.4.42 HwtLpCsEnB

- **Description:** HwtLpCsEnB: Programmed by PHY training firmware.
- **Size:** 2 bits
- **Offset:** 0x70000+0x73
- **Exists:** Always

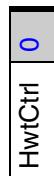


Table 13-378 Fields for Register: HwtLpCsEnB

Bits	Name	Memory Access	Description
1:0	HwtLpCsEnB	R/W	<p>HwtLpCsEnB: Programmed by PHY training firmware.</p> <p><b>Value After Reset:</b> 0x3</p> <p><b>Exists:</b> Always</p>

### 13.4.43 HwtCtrl

- **Description:** HwtCtrl: Programmed by PHY training firmware to support LPDDR4X/5 DRAM drift compensation.
- **Size:** 1 bit
- **Offset:** 0x70000+0x77
- **Exists:** Always



**Table 13-379 Fields for Register: HwtCtrl**

Bits	Name	Memory Access	Description
0	HwtCtrl	R/W	<p>HwtCtrl: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

### 13.4.44 HwtControlOvr

- **Description:** HwtControlOvr: Reserved for PHY training firmware use.
- **Size:** 12 bits
- **Offset:** 0x70000+0x7a
- **Exists:** Always



Table 13-380 Fields for Register: HwtControlOvr

Bits	Name	Memory Access	Description
11:0	HwtControlOvr	R/W	HwtControlOvr: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.45 ScratchPadPPGC

- **Description:** ScratchPadPPGC: ScratchPad for PPGC
- **Size:** 16 bits
- **Offset:** 0x70000+0x7d
- **Exists:** Always



**Table 13-381 Fields for Register: ScratchPadPPGC**

Bits	Name	Memory Access	Description
15:0	ScratchPadPPGC	R/W	<p>ScratchPadPPGC: ScratchPad for PPGC. As required for Automotive. In HMZCAL, When ZcalStopClk=1, this csr can not be accessed.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.46 HwtControlVal

- **Description:** HwtControlVal: Reserved for PHY training firmware use.
- **Size:** 12 bits
- **Offset:** 0x70000+0x7e
- **Exists:** Always



**Table 13-382 Fields for Register: HwtControlVal**

Bits	Name	Memory Access	Description
11:0	HwtControlVal	R/W	<p>HwtControlVal: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.47 ForceHWTClkGaterEnables

- **Name:** Forces internal clock gaters.
- **Description:** ForceHWTClkGaterEnables: Forces internal clock enables High/Low
- **Size:** 4 bits
- **Offset:** 0x70000+0x80
- **Exists:** Always



**Table 13-383 Fields for Register: ForceHWTClkGaterEnables**

Bits	Name	Memory Access	Description
3	ForcePIEClkEnLow	R/W	ForcePIEClkEnLow: Forces the PIE Clk enable to be 0. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
2	ForcePIEClkEnHigh	R/W	ForcePIEClkEnHigh: Forces the PIE Clk enable to be 1. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
1	ForceACSMClkEnLow	R/W	ForceACSMClkEnLow: Forces the ACSM Clk enable to be 0. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
0	ForceACSMClkEnHigh	R/W	ForceACSMClkEnHigh: Forces the ACSM Clk enable to be 1. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.48 MasUpdGoodCtr

- **Description:** MasUpdGoodCtr: Counts successful PHY Master Interface Updates (PPTs)
- **Size:** 16 bits
- **Offset:** 0x70000+0xb5
- **Exists:** Always



**Table 13-384 Fields for Register: MasUpdGoodCtr**

Bits	Name	Memory Access	Description
15:0	MasUpdGoodCtr	R	<p>MasUpdGoodCtr: This register increments whenever the Memory Controller acknowledges a PHY Master Interface request (i.e., a request for the PHY to take over the DFI for Periodic Phase Training (PPT)) within the valid response interval. In a 2-channel system, both MC channels must ack. The counter saturates at max value 65535.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.4.49 PhyUpd0GoodCtr

- **Description:** PhyUpd0GoodCtr: Counts acknowledged PHY-initiated DFI0 Interface Updates
- **Size:** 16 bits
- **Offset:** 0x70000+0xb6
- **Exists:** Always



**Table 13-385 Fields for Register: PhyUpd0GoodCtr**

Bits	Name	Memory Access	Description
15:0	PhyUpd0GoodCtr	R	<p>PhyUpd0GoodCtr: This register increments whenever the Memory Controller acknowledges a PHY-initiated DFI0 interface update request. The counter saturates at max value 65535.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.4.50 PhyUpd1GoodCtr

- **Description:** PhyUpd1GoodCtr: Counts acknowledged PHY-initiated DFI1 Interface Updates
- **Size:** 16 bits
- **Offset:** 0x70000+0xb7
- **Exists:** Always



**Table 13-386 Fields for Register: PhyUpd1GoodCtr**

Bits	Name	Memory Access	Description
15:0	PhyUpd1GoodCtr	R	<p>PhyUpd1GoodCtr: This register increments whenever the Memory Controller acknowledges a PHY-initiated DFI1 interface update request. The counter saturates at max value 65535.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.4.51 CtlUpd0GoodCtr

- **Description:** CtlUpd0GoodCtr: Counts acknowledged Memory Controller DFI0 Interface Updates
- **Size:** 16 bits
- **Offset:** 0x70000+0xb8
- **Exists:** Always



**Table 13-387 Fields for Register: CtlUpd0GoodCtr**

Bits	Name	Memory Access	Description
15:0	CtlUpd0GoodCtr	R	<p>CtlUpd0GoodCtr: This register increments whenever the PHY acknowledges a Memory Controller-initiated DFI0 interface update request. The counter saturates at max value 65535.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.4.52 CtlUpd1GoodCtr

- **Description:** CtlUpd1GoodCtr: Counts acknowledged Memory Controller DFI1 Interface Updates
- **Size:** 16 bits
- **Offset:** 0x70000+0xb9
- **Exists:** Always



**Table 13-388 Fields for Register: CtlUpd1GoodCtr**

Bits	Name	Memory Access	Description
15:0	CtlUpd1GoodCtr	R	<p>CtlUpd1GoodCtr: This register increments whenever the PHY acknowledges a. Memory Controller-initiated DFI1 interface update request. The counter saturates at max value 65535.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.4.53 MasUpdFailCtr

- **Description:** MasUpdFailCtr: Counts unsuccessful PHY Master Interface Updates
- **Size:** 16 bits
- **Offset:** 0x70000+0xba
- **Exists:** Always



**Table 13-389 Fields for Register: MasUpdFailCtr**

Bits	Name	Memory Access	Description
15:0	MasUpdFailCtr	R	<p>MasUpdFailCtr: This register increments whenever the PHY asserts a PHY Master Interface request, but the Memory Controller does not acknowledge the request within the allowed interval, such that dfi_error is asserted. The counter saturates at max value 65535. In a 2-channel system, a failure occurs if either MC channels fail to ack.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.4.54 PhyUpd0FailCtr

- **Description:** PhyUpd0FailCtr: Counts unsuccessful PHY-initiated DFI0 Interface Updates
- **Size:** 16 bits
- **Offset:** 0x70000+0xbb
- **Exists:** Always



**Table 13-390 Fields for Register: PhyUpd0FailCtr**

Bits	Name	Memory Access	Description
15:0	PhyUpd0FailCtr	R	<p>PhyUpd0FailCtr: This register increments whenever the PHY asserts a DFI0. Interface update request, but the Memory Controller does not acknowledge the request within the allowed interval, such that dfi_error is asserted. The counter saturates at max value 65535.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.4.55 PhyUpd1FailCtr

- **Description:** PhyUpd1FailCtr: Counts unsuccessful PHY-initiated DFI1 Interface Updates
- **Size:** 16 bits
- **Offset:** 0x70000+0xbc
- **Exists:** Always



**Table 13-391 Fields for Register: PhyUpd1FailCtr**

Bits	Name	Memory Access	Description
15:0	PhyUpd1FailCtr	R	<p>PhyUpd1FailCtr: This register increments whenever the PHY asserts a DFI1. Interface update request, but the Memory Controller does not acknowledge the request within the allowed interval, such that dfi_error is asserted. The counter saturates at max value 65535.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.4.56 PhyPerfCtrEnable

- **Name:** Enables for Performance Counters
- **Description:** PhyPerfCtrEnable: Each bit corresponds to one performance counter. Set the bit to enable the counter. Clearing the bit stops and clears the counter.
- **Size:** 8 bits
- **Offset:** 0x70000+0xbd
- **Exists:** Always

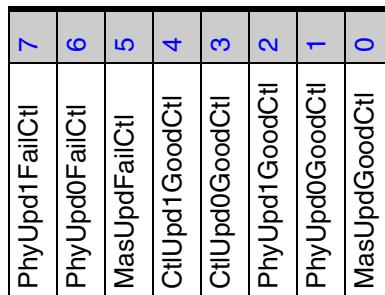


Table 13-392 Fields for Register: PhyPerfCtrEnable

Bits	Name	Memory Access	Description
7	PhyUpd1FailCtl	R/W	<p>PhyUpd1FailCtl: Enables PhyUpd1FailCtl  <b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
6	PhyUpd0FailCtl	R/W	<p>PhyUpd0FailCtl: Enables PhyUpd0FailCtl  <b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
5	MasUpdFailCtl	R/W	<p>MasUpdFailCtl: Enables MasUpdFailCtl  <b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
4	CtlUpd1GoodCtl	R/W	<p>CtlUpd1GoodCtl: Enables CtlUpd1GoodCtl  <b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
3	CtlUpd0GoodCtl	R/W	<p>CtlUpd0GoodCtl: Enables CtlUpd0GoodCtl  <b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
2	PhyUpd1GoodCtl	R/W	<p>PhyUpd1GoodCtl: Enables PhyUpd1GoodCtl  <b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
1	PhyUpd0GoodCtl	R/W	PhyUpd0GoodCtl: Enables PhyUpd0GoodCtr <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
0	MasUpdGoodCtl	R/W	MasUpdGoodCtl: Enables MasUpdGoodCtr <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.57 DfiHandshakeDelays1\_pX (for X = 0; X <= 3)

- **Name:** Small delays on handshake signals per AC Channel
- **Description:** DfiHandshakeDelays1\_pX: Add assertion/deassertion delays on handshake signals. Logic assumes that dfi signal assertions exceed the programmed delays
- **Size:** 12 bits
- **Offset:** 0x70000+0xe6+(X\*0x100000)
- **Exists:** Always

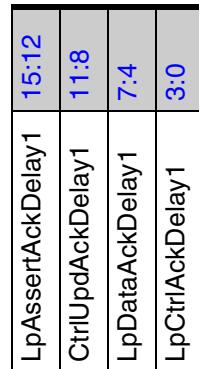


**Table 13-393 Fields for Register: DfiHandshakeDelays1\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
11:8	CtrlUpdReqDelay1	R/W	<p>CtrlUpdReqDelay1: Adds 0-15 DfiClks of additional delay after dfi_ctrlupd_req asserts, before the PHY takes any action (such as starting DDL calibration).</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
7:4	PhyUpdReqDelay1	R/W	<p>PhyUpdReqDelay1: For Debug Use Only. Must be 0. Adds 0-15 DfiClks of additional delay after the PHY completes all PHY update activities, before de-asserting dfi_phyupd_req.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
3:0	PhyUpdAckDelay1	R/W	<p>PhyUpdAckDelay1: For Debug Use Only. Must be 0. Adds 0-15 DfiClks of additional delay after dfi_phyupd_ack asserts, before the PHY takes any action (such as starting DDL calibration).</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.58 DfiRespHandshakeDelays1\_pX (for X = 0; X <= 3)

- **Name:** Small delays on handshake signals per AC Channel
- **Description:** DfiRespHandshakeDelays1\_pX: Add assertion/deassertion delays on handshake signals. Logic assumes that dfi signal assertions exceed the programmed delays
- **Size:** 16 bits
- **Offset:** 0x70000+0xeb+(X\*0x100000)
- **Exists:** Always

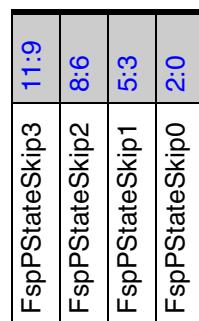


**Table 13-394 Fields for Register: DfiRespHandshakeDelays1\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:12	LpAssertAckDelay1	R/W	<p>LpAssertAckDelay1: For Debug Use Only. Must be 0. Adds 0-15 DfiClks of additional delay before the PHY would normally assert dfi_lp_data_ack.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
11:8	CtrlUpdAckDelay1	R/W	<p>CtrlUpdAckDelay1: For Debug Use Only. Must be 0. Adds 0-15 DfiClks of additional delay after the PHY completes all PHY update activities, before de-asserting dfi_ctrlupd_ack.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
7:4	LpDataAckDelay1	R/W	<p>LpDataAckDelay1: Adds 0-15 DfiClks of additional delay after the PHY would normally deassert dfi_lp_data_ack.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
3:0	LpCtrlAckDelay1	R/W	<p>LpCtrlAckDelay1: Adds 0-15 DfiClks of additional delay after the PHY would normally deassert dfi_lp_ctrl_ack.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.59 FspSkipList

- **Name:** FspState update skip list.
- **Description:** FspSkipList: Used by the FSP tracking logic to skip updating the FspState register for specified values of destPState  
This register is programmed by PhyInit. Values depend on the PIE code image used. The values programmed indicates destPstate[2:0] values that do not change the DRAM FSP.
- **Size:** 12 bits
- **Offset:** 0x70000+0xf0
- **Exists:** Always



**Table 13-395 Fields for Register: FspSkipList**

Bits	Name	Memory Access	Description
11:9	FspPStateSkip3	R/W	FspPStateSkip3: This destPstate should skip FspState update. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
8:6	FspPStateSkip2	R/W	FspPStateSkip2: This destPstate should skip FspState update. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
5:3	FspPStateSkip1	R/W	FspPStateSkip1: This destPstate should skip FspState update. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
2:0	FspPStateSkip0	R/W	FspPStateSkip0: This destPstate should skip FspState update. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.60 PPGCReservedX (for X == 0)

- **Description:** PPGCReservedX: Reserved for future use
- **Size:** 16 bits
- **Offset:** 0x70000+0xfa
- **Exists:** Always



**Table 13-396 Fields for Register: PPGCReservedX (for X == 0)**

Bits	Name	Memory Access	Description
15:0	PPGCReserved0	R/W	PPGCReserved0: Reserved for future use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.61 PUBReservedP1\_pX (PPGC) (for X = 0; X <= 3)

- **Description:** PUBReservedP1\_pX: Reserved for future use
- **Size:** 8 bits
- **Offset:** 0x70000+0xff+(X\*0x100000)
- **Exists:** Always



**Table 13-397 Fields for Register: PUBReservedP1\_pX (PPGC) (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
7:0	PUBReservedP1_p0	R/W	PUBReservedP1_p0: Reserved for future use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.62 PhyInterruptOverride

- **Description:** PhyInterruptOverride: Interrupt Override Debug Register
- **Size:** 16 bits
- **Offset:** 0x70000+0x11a
- **Exists:** Always

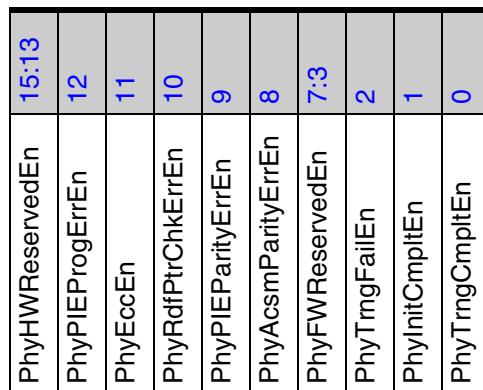


**Table 13-398 Fields for Register: PhyInterruptOverride**

Bits	Name	Memory Access	Description
15:0	PhyInterruptOverride	R/W	<p>PhyInterruptOverride: This register is used for debug. A 0 -&gt; 1 transition on any bit causes the corresponding interrupt in csrPhyInterruptStatus to be triggered, if the interrupt is masked and enabled. Hardware-generated interrupts will be ignored when the corresponding bit is 1 in this register. This permits software emulation of HW interrupts.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.63 PhyInterruptEnable

- **Name:** Interrupt Enable Bits
- **Description:** PhyInterruptEnable: Enables various interrupts to appear in the PhyInterruptStatus CSR. Only interrupts which are enabled by writing a 1 to the appropriate location in this CSR will be observable in the PhyInterruptStatus CSR. If a 0 is written in a particular location the corresponding interrupt will not be visible in the PhyInterruptStatus CSR. See the description of PhyInterruptStatus for more information on specific interrupts.
- **Size:** 16 bits
- **Offset:** 0x70000+0x11b
- **Exists:** Always



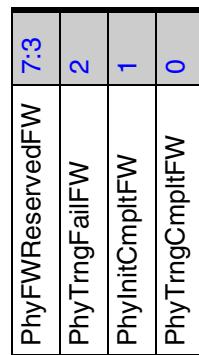
**Table 13-399 Fields for Register: PhyInterruptEnable**

Bits	Name	Memory Access	Description
15:13	PhyHWReservedEn	R/W	PhyHWReservedEn: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
12	PhyPIEPProgErrEn	R/W	PhyPIEPProgErrEn: Enable for the PIE Programming Error Interrupt <ul style="list-style-type: none"> <li>■ 0 : Interrupt not enabled</li> <li>■ 1 : Interrupt enabled</li> </ul> <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
11	PhyEccEn	R/W	PhyEccEn: Enable for the ARC ECC Interrupt <ul style="list-style-type: none"> <li>■ 0 : Interrupt not enabled</li> <li>■ 1 : Interrupt enabled</li> </ul> <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

Bits	Name	Memory Access	Description
10	PhyRdfPtrChkErrEn	R/W	<p>PhyRdfPtrChkErrEn: Enable for RdfPtrChkErr Interrupt</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not enabled</li> <li>■ 1 : Interrupt enabled</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
9	PhyPIEParityErrEn	R/W	<p>PhyPIEParityErrEn: Enable for the PIE Parity Error Interrupt</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not enabled</li> <li>■ 1 : Interrupt enabled</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
8	PhyAcsmParityErrEn	R/W	<p>PhyAcsmParityErrEn: Enable for the ACSM Parity Error Interrupt</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not enabled</li> <li>■ 1 : Interrupt enabled</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
7:3	PhyFWReservedEn	R/W	<p>PhyFWReservedEn: Reserved</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
2	PhyTrngFailEn	R/W	<p>PhyTrngFailEn: Enable for the PHY Training Failure interrupt.</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not enabled</li> <li>■ 1 : Interrupt enabled</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
1	PhyInitCmpltEn	R/W	<p>PhyInitCmpltEn: Enable for the PHY Initialization Complete interrupt.</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not enabled</li> <li>■ 1 : Interrupt enabled</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
0	PhyTrngCmpltEn	R/W	<p>PhyTrngCmpltEn: Enable for the PHY Training Complete interrupt.</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not enabled</li> <li>■ 1 : Interrupt enabled</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.64 PhyInterruptFWControl

- **Name:** Interrupt Firmware Control Bits
- **Description:** PhyInterruptFWControl: This register is written ONLY by the Firmware. A 0 to 1 transition for any of the interrupts will set the corresponding bit location in the PhyInterruptStatus CSR if the corresponding bit location in the PhyInterruptEnable CSR is set to 1. See the description of PhyInterruptStatus for more information on specific interrupts.
- **Size:** 8 bits
- **Offset:** 0x70000+0x11c
- **Exists:** Always



**Table 13-400 Fields for Register: PhyInterruptFWControl**

Bits	Name	Memory Access	Description
7:3	PhyFWReservedFW	R/W	<p>PhyFWReservedFW: Reserved</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
2	PhyTrngFailFW	R/W	<p>PhyTrngFailFW: PHY Training Failure Firmware interrupt.</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not asserted</li> <li>■ 1 : Interrupt asserted</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
1	PhyInitCmpltFW	R/W	<p>PhyInitCmpltFW: PHY Initialization Complete Firmware interrupt.</p> <p><b>Note:</b> Use of this interrupt is optional. The PHY does not set this interrupt. In the event that there are multiple processors or threads in the SOC, the thread loading PHYINIT can write this bit as part of. PhyInit userCustomPost() or manually via APB after first dfi_init_start.</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not asserted</li> <li>■ 1 : Interrupt asserted</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
0	PhyTrngCmpltFW	R/W	<p>PhyTrngCmpltFW: PHY Training Complete Firmware interrupt.</p> <ul style="list-style-type: none"><li>■ 0 : Interrupt not asserted</li><li>■ 1 : Interrupt asserted</li></ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.65 PhyInterruptMask

- **Name:** Interrupt Mask Bits
- **Description:** PhyInterruptMask: This register is used to mask various interrupts to prevent them from affecting the output pin dwc\_ddrphy\_int\_n. Only interrupts which are not masked by clearing to 0 the corresponding location in this CSR will be able to affect the dwc\_ddrphy\_int\_n output when asserted in the PhyInterruptStatus CSR. If a 1 is written in a particular location that interrupt will still be visible in the PhyInterruptStatus CSR but that interrupt will not affect the output pin dwc\_ddrphy\_int\_n. See the description of PhyInterruptStatus for more information on specific interrupts.
- **Size:** 16 bits
- **Offset:** 0x70000+0x11d
- **Exists:** Always

PhyHWReservedMsk	15:13															
PhyPIEProgErrMsk	12															
PhyEccMsk	11															
PhyRdfPtrChkErrMsk	10															
PhyPIEParityErrMsk	9															
PhyAcsmParityErrMsk	8															
PhyFWReservedMsk	7:3															
PhyTrngFailMsk	2															
PhyInitCmpltMsk	1															
PhyTrngCmpltMsk	0															

Table 13-401 Fields for Register: PhyInterruptMask

Bits	Name	Memory Access	Description
15:13	PhyHWReservedMsk	R/W	<p>PhyHWReservedMsk: Reserved</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
12	PhyPIEProgErrMsk	R/W	<p>PhyPIEProgErrMsk: Mask for the PIE Programming ErrorInterrupt</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not masked</li> <li>■ 1 : Interrupt masked</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
11	PhyEccMsk	R/W	<p>PhyEccMsk: Mask for the ARC ECC Interrupt</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not masked</li> <li>■ 1 : Interrupt masked</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
10	PhyRdfPtrChkErrMsk	R/W	<p>PhyRdfPtrChkErrMsk: Mask for the RdfPtrChkErr</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not masked</li> <li>■ 1 : Interrupt masked</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
9	PhyPIEParityErrMsk	R/W	<p>PhyPIEParityErrMsk: Mask for the PIE Parity Error Interrupt</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not masked</li> <li>■ 1 : Interrupt masked</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
8	PhyAcsmParityErrMsk	R/W	<p>PhyAcsmParityErrMsk: Mask for the ACSM Parity Error Interrupt</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not masked</li> <li>■ 1 : Interrupt masked</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
7:3	PhyFWReservedMsk	R/W	<p>PhyFWReservedMsk: Reserved</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
2	PhyTrngFailMsk	R/W	<p>PhyTrngFailMsk: Mask for the PHY Training Failure interrupt.</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not masked</li> <li>■ 1 : Interrupt masked</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
1	PhyInitCmpltMsk	R/W	<p>PhyInitCmpltMsk: Mask for the PHY Initialization Complete interrupt.</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not masked</li> <li>■ 1 : Interrupt masked</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
0	PhyTrngCmpltMsk	R/W	<p>PhyTrngCmpltMsk: Mask for the PHY Training Complete interrupt.</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not masked</li> <li>■ 1 : Interrupt masked</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.66 PhyInterruptClear

- **Name:** Interrupt Clear Bits
- **Description:** PhyInterruptClear: This register clears various interrupts and will negate the output pin dwc\_ddrphy\_int\_n if the cleared interrupt was the cause of the assertion of the output pin dwc\_d-drphy\_int\_n. A 0 to 1 transition on a bit will clear the corresponding interrupt, and cause the hardware to immediately clear the bit. See the description of PhyInterruptStatus for more information on specific interrupts.
- **Size:** 16 bits
- **Offset:** 0x70000+0x11e
- **Exists:** Always

PhyHWReservedClr	15:13
PhyPIEPProgErrClr	12
PhyEccClr	11
PhyRdfPrChkErrClr	10
PhyPIEParityErrClr	9
PhyAcsmparityErrClr	8
PhyFWReservedClr	7:3
PhyTrngFailClr	2
PhyInitCmpltClr	1
PhyTrngCmpltClr	0

Table 13-402 Fields for Register: PhyInterruptClear

Bits	Name	Memory Access	Description
15:13	PhyHWReservedClr	R/W	PhyHWReservedClr: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
12	PhyPIEPProgErrClr	R/W	PhyPIEPProgErrClr: Clear for the PHY Programming Error Interrupt <ul style="list-style-type: none"> <li>■ 0 : Interrupt not affected</li> <li>■ 1 : Interrupt cleared</li> </ul> <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
11	PhyEccClr	R/W	PhyEccClr: Clear for the ARC ECC Interrupt <ul style="list-style-type: none"> <li>■ 0 : Interrupt not affected</li> <li>■ 1 : Interrupt cleared</li> </ul> <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

Bits	Name	Memory Access	Description
10	PhyRdfPtrChkErrClr	R/W	<p>PhyRdfPtrChkErrClr: Clear for the RdfPtrChkErr</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not affected</li> <li>■ 1 : Interrupt cleared</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
9	PhyPIEParityErrClr	R/W	<p>PhyPIEParityErrClr: Clear for the PIE Parity Error Interrupt</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not affected</li> <li>■ 1 : Interrupt cleared</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
8	PhyAcsmParityErrClr	R/W	<p>PhyAcsmParityErrClr: Clear for the ACSM Parity Error Interrupt</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not affected</li> <li>■ 1 : Interrupt cleared</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
7:3	PhyFWReservedClr	R/W	<p>PhyFWReservedClr: Reserved</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
2	PhyTrngFailClr	R/W	<p>PhyTrngFailClr: Clear for the PHY Training Failure interrupt.</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not affected</li> <li>■ 1 : Interrupt cleared</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
1	PhyInitCmpltClr	R/W	<p>PhyInitCmpltClr: Clear for the PHY Initialization Complete interrupt.</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not affected</li> <li>■ 1 : Interrupt cleared</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
0	PhyTrngCmpltClr	R/W	<p>PhyTrngCmpltClr: Clear for the PHY Training Complete interrupt.</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not affected</li> <li>■ 1 : Interrupt cleared</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.67 PhyInterruptStatus

- **Name:** Interrupt Status Bits
- **Description:** PhyInterruptStatus: This register displays the current status of various PHY interrupts. This is a read only CSR. The PHY is the only entity which can write to it. In order for a bit to become set in the PhyInterruptStatus CSR the corresponding bit must be set in the PhyInterruptEnable CSR. If an interrupt is set in the PhyInterruptStatus CSR and the corresponding bit in the PhyInterruptMask CSR is cleared the dwc\_ddrphy\_int\_n output signal will be asserted. Once the dwc\_ddrphy\_int\_n output signal is asserted it can only be negated when all unmasked interrupts in the PhyInterruptStatus CSR have been cleared by writing the corresponding bit in the PhyInterruptClear CSR to 1.
- **Size:** 16 bits
- **Offset:** 0x70000+0x11f
- **Exists:** Always

PhyHWReserved	15:13
PhyPIEProgErr	12
PhyEccErr	11
PhyRdfPrcChkErr	10
PhyPIEParityErr	9
PhyAcsmparityErr	8
PhyFWReserved	7:3
PhyTrngFail	2
PhyInitCmplt	1
PhyTrngCmplt	0

Table 13-403 Fields for Register: PhyInterruptStatus

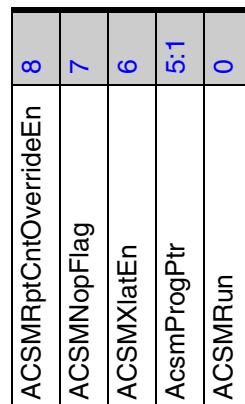
Bits	Name	Memory Access	Description
15:13	PhyHWReserved	R	<p>PhyHWReserved: Reserved  <b>Value After Reset:</b> 0x0  <b>Exists:</b> Always  <b>Reset Mask:</b> 0x0  <b>Volatile:</b> true</p>
12	PhyPIEProgErr	R	<p>PhyPIEProgErr: PIE Programming Error - see PIEProgErrStatus CSR to see specifics of the error</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not enabled or not asserted</li> <li>■ 1 : Interrupt enabled and asserted</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always  <b>Reset Mask:</b> 0x0  <b>Volatile:</b> true</p>

Bits	Name	Memory Access	Description
11	PhyEccErr	R	<p>PhyEccErr: ARC ECC Interrupt - see csrArcEccIndications for specifics</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not enabled or not asserted</li> <li>■ 1 : Interrupt enabled and asserted</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
10	PhyRdfPtrChkErr	R	<p>PhyRdfPtrChkErr: RdfPtrChkErr Interrupt. Indicates error in read data fifo pointers when fifo should be empty/idle.</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not enabled and asserted</li> <li>■ 1 : Interrupt enabled and asserted</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
9	PhyPIEParityErr	R	<p>PhyPIEParityErr: PIE Parity Error Interrupt</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not enabled and asserted</li> <li>■ 1 : Interrupt enabled and asserted</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
8	PhyAcsmParityErr	R	<p>PhyAcsmParityErr: ACSM Parity Error Interrupt</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not enabled and asserted</li> <li>■ 1 : Interrupt enabled and asserted</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
7:3	PhyFWReserved	R	<p>PhyFWReserved: Reserved</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

Bits	Name	Memory Access	Description
2	PhyTrngFail	R	<p>PhyTrngFail: PHY Training Failure interrupt. Assertion of this interrupt indicates there was an issue during training which did not allow the PHY to successfully complete training.</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not enabled and asserted</li> <li>■ 1 : Interrupt enabled and asserted</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
1	PhyInitCmplt	R	<p>PhyInitCmplt: PHY Initialization Complete interrupt. Assertion of this interrupt indicates that the PHY Initialization Engine has been fully loaded and the PHY is ready and able to assert dfi_init_complete following the assertion of dfi_init_start.</p> <p><b>Note:</b> Use of this interrupt is optional. The PHY does not set this interrupt. In the event that there are multiple processors or threads in the SOC, the thread loading PHYINIT can write this bit as part of. PhyInit userCustomPost() or manually via APB after first dfi_init_start.</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not enabled and asserted</li> <li>■ 1 : Interrupt enabled and asserted</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
0	PhyTrngCmplt	R	<p>PhyTrngCmplt: PHY Training Complete interrupt. Assertion of this interrupt indicates that the PHY has successfully trained the memory interface at all indicated frequencies and is ready for the PHY Initialization Engine to be loaded.</p> <ul style="list-style-type: none"> <li>■ 0 : Interrupt not enabled and asserted</li> <li>■ 1 : Interrupt enabled and asserted</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.4.68 ACSMRunCtrl

- **Name:** Enable for ACSMs and optionally specifies the AcsmProgPtr .
- **Description:** ACSMRunCtrl: This register contains the set of runtime control for the ACSM. AcsmRun starts the ACSM and AcsmProgPtr specifies which ACSM program to execute of the AcsmPtrXlat table is used. The AcsmPtrXlatEn is disabled. AcsmProgPtr is don't care.
- **Size:** 9 bits
- **Offset:** 0x70000+0x120
- **Exists:** Always



**Table 13-404 Fields for Register: ACSMRunCtrl**

Bits	Name	Memory Access	Description
8	ACSMRptCntOverrideEn	R/W	ACSMRptCntOverrideEn: Enables ACSMRptCntOverride feature When this field is 1, ACSMRptCntOverride must be non-zero. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	ACSMNopFlag	R/W	ACSMNopFlag: Temporarily disables address de-referencing given the SkipMrw internal signal. When set to 1, PHY will always issue MRW's when switching PStates. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6	ACSMXlatEn	R/W	ACSMXlatEn: When set to 1, de-referencing is enabled. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
5:1	AcsmProgPtr	R/W	AcsmProgPtr: Used for indexing AcsmPtrXlat registers. Programmed by PIE during frequency changes and should not be written by user. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

Bits	Name	Memory Access	Description
0	ACSMRun	R/W	<p>ACSMRun: This is the start bit for the playback engine. The ACSM starts when it detects a 0-&gt;1 transition on the bit. When the Run bit is set and the Done bit is not set, the ACSM is running.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.69 ACSMDone

- **Description:** ACSMDone: Done Status for ACSMs
- **Size:** 1 bit
- **Offset:** 0x70000+0x121
- **Exists:** Always



**Table 13-405 Fields for Register: ACSMDone**

Bits	Name	Memory Access	Description
0	ACSMDone	R	<p>ACSMDone: This is the done bit for the playback engine. It is set when the ACSM has completed its sequence. When the Run bit is set and the Done bit is not set, the ACSM is running.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.4.70 ACSMStartAddr\_pX (for X = 0; X <= 3)

- **Description:** ACSMStartAddr\_pX: Start Address for ACSM Sequence
- **Size:** 11 bits
- **Offset:** 0x70000+0x122+(X\*0x100000)
- **Exists:** Always



**Table 13-406 Fields for Register: ACSMStartAddr\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
10:0	ACSMStartAddr_p0	R/W	<p>ACSMStartAddr_p0: Allows user to change the first executed address. Bit 0 is ignored and assumed to be 0.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.71 ACSMStopAddr\_pX (for X = 0; X <= 3)

- **Description:** ACSMStopAddr\_pX: Last address of Playback Sequence
- **Size:** 11 bits
- **Offset:** 0x70000+0x123+(X\*0x100000)
- **Exists:** Always



**Table 13-407 Fields for Register: ACSMStopAddr\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
10:0	ACSMStopAddr_p0	R/W	ACSMStopAddr_p0: Indicates the last address to be executed by the corresponding ACSM. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.72 ACSMLastAddr

- **Description:** ACSMLastAddr: Last executed address of Playback Sequence
- **Size:** 11 bits
- **Offset:** 0x70000+0x124
- **Exists:** Always



**Table 13-408 Fields for Register: ACSMLastAddr**

Bits	Name	Memory Access	Description
10:0	ACSMLastAddr	R	ACSMLastAddr: Indicates the last executed address by ACSM. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always <b>Reset Mask:</b> 0x0 <b>Volatile:</b> true

### 13.4.73 ACSMAlgIncVal

- **Description:** ACSMAlgIncVal: Increment value for ACSM Algorithmic Address Generator
- **Size:** 14 bits
- **Offset:** 0x70000+0x125
- **Exists:** Always



**Table 13-409 Fields for Register: ACSMAlgIncVal**

Bits	Name	Memory Access	Description
13:0	ACSMAlgIncVal	R/W	<p>ACSMAlgIncVal: This value is added to an internal counter which is then added to the CA Address generated by ACSM. When the ACSMAddressMask is non-zero, a bit location set to one in that register should not be considered active in this register. So, for example, if the AddressMask is 0x2, and user wants to specify a value of 0x3, user would specify 0x5 (ignoring bit 1).</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.74 ACSMAddressMask

- **Description:** ACSMAddressMask: CA Address Mask
- **Size:** 14 bits
- **Offset:** 0x70000+0x126
- **Exists:** Always



**Table 13-410 Fields for Register: ACSMAddressMask**

Bits	Name	Memory Access	Description
13:0	ACSMAddressMask	R/W	ACSMAddressMask: A one indicates that the corresponding bit in CA[13:0] is not part of the address increment calculation. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.75 ACSMOuterLoopRepeatCnt

- **Description:** ACSMOuterLoopRepeatCnt: Outer Loop Repeat
- **Size:** 16 bits
- **Offset:** 0x70000+0x127
- **Exists:** Always

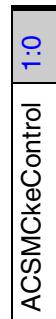


**Table 13-411 Fields for Register: ACSMOuterLoopRepeatCnt**

Bits	Name	Memory Access	Description
15:0	ACSMOuterLoopRepeatCnt	R/W	<p>ACSMOuterLoopRepeatCnt: The repeat count for the outer loop for ACSM Channel. A repeat count of n means that the code in the loop is executed n+1 times</p> <ul style="list-style-type: none"> <li>■ When csrACSMInfiniteOLRC is set, the outer loop repeats forever if csrACSMRun is set.</li> <li>■ When csrACSMInfiniteOLRC is cleared, the loop counter is set to 0 and the control flow exits the loop.</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.4.76 ACSMCkeControl

- **Description:** ACSMCkeControl: CKE Control for ACSM
- **Size:** 2 bits
- **Offset:** 0x70000+0x128
- **Exists:** Always

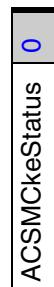


**Table 13-412 Fields for Register: ACSMCkeControl**

Bits	Name	Memory Access	Description
1:0	ACSMCkeControl	R/W	<p>ACSMCkeControl: Provides Control for CKE Bits. CKE can be forced to 0, forced to 1, or toggled by the CKE field in the ACSM Control Word.</p> <ul style="list-style-type: none"> <li>■ 00 = NOP (retain prior value)</li> <li>■ 01 = Controlled by ACSM</li> <li>■ 10 = Force 0</li> <li>■ 11 = Force 1</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.4.77 ACSMCkeStatus

- **Description:** ACSMCkeStatus: CKE Status for ACSM
- **Size:** 1 bit
- **Offset:** 0x70000+0x129
- **Exists:** Always

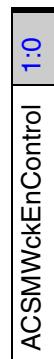


**Table 13-413 Fields for Register: ACSMCkeStatus**

Bits	Name	Memory Access	Description
0	ACSMCkeStatus	R	<p>ACSMCkeStatus: Shows the ACSM-derived value for the CKE bit</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.4.78 ACSMWckEnControl

- **Description:** ACSMWckEnControl: hwt\_wck\_en Control for ACSM
- **Size:** 2 bits
- **Offset:** 0x70000+0x12a
- **Exists:** Always

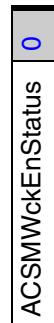


**Table 13-414 Fields for Register: ACSMWckEnControl**

Bits	Name	Memory Access	Description
1:0	ACSMWckEnControl	R/W	<p>ACSMWckEnControl: Provides Control for hwt_wck_en.</p> <ul style="list-style-type: none"> <li>■ 00 = NOP</li> <li>■ 01 = Controlled by ACSM (setting WCKTG causes a toggle)</li> <li>■ 10 = Force 0, 11 = Force 1</li> <li>■ 11 = Force 1</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.79 ACSMWckEnStatus

- **Description:** ACSMWckEnStatus: wck Status for ACSM
- **Size:** 1 bit
- **Offset:** 0x70000+0x12b
- **Exists:** Always



**Table 13-415 Fields for Register: ACSMWckEnStatus**

Bits	Name	Memory Access	Description
0	ACSMWckEnStatus	R	ACSMWckEnStatus: Value of internal wck tracking flop - for internal debug only <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always <b>Reset Mask:</b> 0x0 <b>Volatile:</b> true

### 13.4.80 ACSMRxEnPulse\_pX (for X = 0; X <= 3)

- **Name:** Timing Parameters for RxEn Pulse
- **Description:** ACSMRxEnPulse\_pX: Contains the delay and pulse width for RxEn
- **Size:** 14 bits
- **Offset:** 0x70000+0x12c+(X\*0x100000)
- **Exists:** Always



**Table 13-416 Fields for Register: ACSMRxEnPulse\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
13:8	ACSMRxEnWidth	R/W	ACSMRxEnWidth: The number of MEMCLKs the RxEn bit is asserted <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	ACSMRxEnDelayReserved	R/W	ACSMRxEnDelayReserved: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6:0	ACSMRxEnDelay	R/W	ACSMRxEnDelay: The number of MEMCLKs between the cycle that the RxEn bit is set in the ACSM and the cycle that RxEn is asserted. Max=95 <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.81 ACSMRxValPulse\_pX (for X = 0; X <= 3)

- **Name:** Timing Parameters for RxVal Pulse
- **Description:** ACSMRxValPulse\_pX: Contains the delay and pulse width for RxVal
- **Size:** 14 bits
- **Offset:** 0x70000+0x12d+(X\*0x100000)
- **Exists:** Always



**Table 13-417 Fields for Register: ACSMRxValPulse\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
13:8	ACSMRxValWidth	R/W	ACSMRxValWidth: The number of MEMCLKs the RxVal bit is asserted <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	ACSMRxValDelayReserved	R/W	ACSMRxValDelayReserved: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6:0	ACSMRxValDelay	R/W	ACSMRxValDelay: The number of MEMCLKs between the cycle that the RxVal bit is set in the ACSM and the cycle that RxVal is asserted. Max=95 <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.82 ACSMTxEnPulse\_pX (for X = 0; X <= 3)

- **Name:** Timing Parameters for TxEn Pulse
- **Description:** ACSMTxEnPulse\_pX: Contains the delay and pulse width for TxEn
- **Size:** 14 bits
- **Offset:** 0x70000+0x12e+(X\*0x100000)
- **Exists:** Always



**Table 13-418 Fields for Register: ACSMTxEnPulse\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
13:8	ACSMTxEnWidth	R/W	ACSMTxEnWidth: The number of MEMCLKs the TxEn bit is asserted <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	ACSMTxEnDelayReserved	R/W	ACSMTxEnDelayReserved: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6:0	ACSMTxEnDelay	R/W	ACSMTxEnDelay: The number of MEMCLKs between the cycle that the TxEn bit is set in the ACSM and the cycle that TxEn is asserted Max=95 <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.83 ACSMWrcsPulse\_pX (for X = 0; X <= 3)

- **Name:** Timing Parameters for Wrcs Pulse
- **Description:** ACSMWrcsPulse\_pX: Contains the delay and pulse width for Wrcs
- **Size:** 14 bits
- **Offset:** 0x70000+0x12f+(X\*0x100000)
- **Exists:** Always



**Table 13-419 Fields for Register: ACSMWrcsPulse\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
13:8	ACSMWrcsWidth	R/W	ACSMWrcsWidth: The number of MEMCLKs the Wrcs bit is asserted <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	ACSMWrcsDelayReserved	R/W	ACSMWrcsDelayReserved: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6:0	ACSMWrcsDelay	R/W	ACSMWrcsDelay: The number of MEMCLKs between the cycle that the TxEn bit is set in the ACSM and the cycle that wrdata_cs_n (as selected by the TG bits) is asserted Max=95 <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.84 ACSMRdcsPulse\_pX (for X = 0; X <= 3)

- **Name:** Timing Parameters for Rdcs Pulse
- **Description:** ACSMRdcsPulse\_pX: Contains the delay and pulse width for Rdcs
- **Size:** 14 bits
- **Offset:** 0x70000+0x130+(X\*0x100000)
- **Exists:** Always



**Table 13-420 Fields for Register: ACSMRdcsPulse\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
13:8	ACSMRdcsWidth	R/W	ACSMRdcsWidth: The number of MEMCLKs the Rdcs bit is asserted <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	ACSMRdcsDelayReserved	R/W	ACSMRdcsDelayReserved: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6:0	ACSMRdcsDelay	R/W	ACSMRdcsDelay: The number of MEMCLKs between the cycle that the RxEn bit is set in the ACSM and the cycle that rddata_cs_n (as selected by the TG bits) is asserted. Max=95 <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.85 ACSMInfiniteOLRC

- **Description:** ACSMInfiniteOLRC: Enables outer loop as infinite for ACSMs
- **Size:** 1 bit
- **Offset:** 0x70000+0x131
- **Exists:** Always



**Table 13-421 Fields for Register: ACSMInfiniteOLRC**

Bits	Name	Memory Access	Description
0	ACSMInfiniteOLRC	R/W	<p>ACSMInfiniteOLRC: This field is programmed as follows:</p> <ul style="list-style-type: none"> <li>■ When set, the outer loop runs forever if csrACSMRun is set.</li> <li>■ When the bit is cleared, the loop counter is set to 0 and the control flow exits the loop.</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.4.86 ACSMDefaultAddr

- **Description:** ACSMDefaultAddr: ca value driven by ACSM when idle
- **Size:** 14 bits
- **Offset:** 0x70000+0x132
- **Exists:** Always



**Table 13-422 Fields for Register: ACSMDefaultAddr**

Bits	Name	Memory Access	Description
13:0	ACSMDefaultAddr	R/W	<p>ACSMDefaultAddr: ca value driven by ACSM when idle</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.87 ACSMDefaultCs

- **Description:** ACSMDefaultCs: cs value driven by ACSM when idle
- **Size:** 2 bits
- **Offset:** 0x70000+0x133
- **Exists:** Always

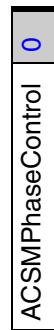


**Table 13-423 Fields for Register: ACSMDefaultCs**

Bits	Name	Memory Access	Description
1:0	ACSMDefaultCs	R/W	ACSMDefaultCs: cs value driven by ACSM when idle <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.88 ACSMStaticCtrl

- **Name:** Control the ACSM Phase mode of operation.
- **Description:** ACSMStaticCtrl: ACSM Static Controls
- **Size:** 1 bit
- **Offset:** 0x70000+0x134
- **Exists:** Always

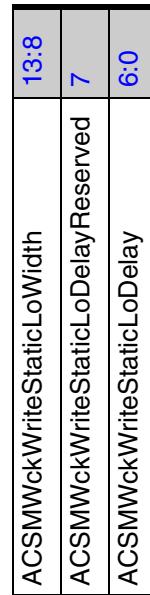


**Table 13-424 Fields for Register: ACSMStaticCtrl**

Bits	Name	Memory Access	Description
0	ACSMPhaseControl	R/W	<p>ACSMPhaseControl: This field is programmed as follows:</p> <ul style="list-style-type: none"> <li>■ 0 = For LP4X, in 1:2 mode, P2/P3 instructions are ignored</li> <li>■ 1 = For LP4X, in 1:2 mode, P2/P3 instructions are processed in a 2nd DfiClk cycle after P0/P1 are processed</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.89 ACSMWckWriteStaticLoPulse\_pX (for X = 0; X <= 3)

- **Name:** Timing Parameters for WckWriteStaticLo Pulse
- **Description:** ACSMWckWriteStaticLoPulse\_pX: Contains the delay and pulse width for WckWriteStaticLo
- **Size:** 14 bits
- **Offset:** 0x70000+0x135+(X\*0x100000)
- **Exists:** Always

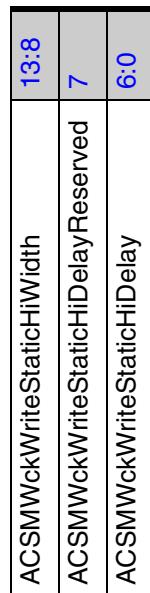


**Table 13-425 Fields for Register: ACSMWckWriteStaticLoPulse\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
13:8	ACSMWckWriteStaticLoWidth	R/W	ACSMWckWriteStaticLoWidth: The number of MEMCLKs hwt_wck_en is asserted <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	ACSMWckWriteStaticLoDelayReseved	R/W	ACSMWckWriteStaticLoDelayReserved: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6:0	ACSMWckWriteStaticLoDelay	R/W	ACSMWckWriteStaticLoDelay: The number of MEMCLKs between the cycle that the WCKWRSLO bit is set in the ACSM and the cycle that hwt_wck_en is asserted. Max=95 <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.90 ACSMWckWriteStaticHiPulse\_pX (for X = 0; X <= 3)

- **Name:** Timing Parameters for WckWriteStaticHi Pulse
- **Description:** ACSMWckWriteStaticHiPulse\_pX: Contains the delay and pulse width for WckWriteStaticHi
- **Size:** 14 bits
- **Offset:** 0x70000+0x136+(X\*0x100000)
- **Exists:** Always



**Table 13-426 Fields for Register: ACSMWckWriteStaticHiPulse\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
13:8	ACSMWckWriteStaticHiWidth	R/W	ACSMWckWriteStaticHiWidth: The number of MEMCLKs hwt_wck_en and hwt_wck_toggle are asserted <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	ACSMWckWriteStaticHiDelayRese rved	R/W	ACSMWckWriteStaticHiDelayReserved: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6:0	ACSMWckWriteStaticHiDelay	R/W	ACSMWckWriteStaticHiDelay: The number of MEMCLKs between the cycle that the WCKWRSHI bit is set in the ACSM and the cycle that hwt_wck_en and hwt_wck_toggle are asserted. hwt_wck_toggle is asserted to 'b01. Max=95 <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.91 ACSMWckWriteTogglePulse\_pX (for X = 0; X <= 3)

- **Name:** Timing Parameters for WckWriteToggle Pulse
- **Description:** ACSMWckWriteTogglePulse\_pX: Contains the delay and pulse width for WckWriteToggle
- **Size:** 14 bits
- **Offset:** 0x70000+0x137+(X\*0x100000)
- **Exists:** Always

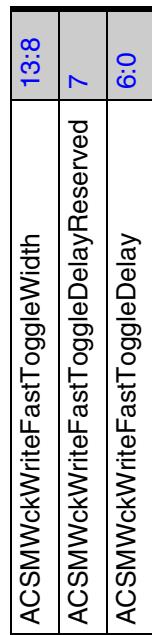


**Table 13-427 Fields for Register: ACSMWckWriteTogglePulse\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
13:8	ACSMWckWriteToggleWidth	R/W	ACSMWckWriteToggleWidth: The number of MEMCLKs hwt_wck_en and hwt_wck_toggle are asserted <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	ACSMWckWriteToggleDelayReserved	R/W	ACSMWckWriteToggleDelayReserved: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6:0	ACSMWckWriteToggleDelay	R/W	ACSMWckWriteToggleDelay: The number of MEMCLKs between the cycle that the WCKWRRTG bit is set in the ACSM and the cycle that hwt_wck_en and hwt_wck_toggle are asserted. hwt_wck_toggle is asserted to 'b10. Max=95 <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.92 ACSMWckWriteFastTogglePulse\_pX (for X = 0; X <= 3)

- **Name:** Timing Parameters for WckWriteFastToggle Pulse
- **Description:** ACSMWckWriteFastTogglePulse\_pX: Contains the delay and pulse width for WckWriteFastToggle
- **Size:** 14 bits
- **Offset:** 0x70000+0x138+(X\*0x100000)
- **Exists:** Always

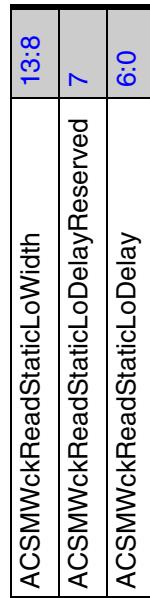


**Table 13-428 Fields for Register: ACSMWckWriteFastTogglePulse\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
13:8	ACSMWckWriteFastToggleWidth	R/W	ACSMWckWriteFastToggleWidth: The number of MEMCLKs hwt_wck_en and hwt_wck_toggle are asserted <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	ACSMWckWriteFastToggleDelayReserved	R/W	ACSMWckWriteFastToggleDelayReserved: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6:0	ACSMWckWriteFastToggleDelay	R/W	ACSMWckWriteFastToggleDelay: The number of MEMCLKs between the cycle that the WCKWRRFTG bit is set in the ACSM and the cycle that hwt_wck_en and hwt_wck_toggle are asserted. hwt_wck_toggle is asserted to 'b11. Max=95 <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.93 ACSMWckReadStaticLoPulse\_pX (for X = 0; X <= 3)

- **Name:** Timing Parameters for WckReadStaticLo Pulse
- **Description:** ACSMWckReadStaticLoPulse\_pX: Contains the delay and pulse width for WckReadStaticLo
- **Size:** 14 bits
- **Offset:** 0x70000+0x139+(X\*0x100000)
- **Exists:** Always

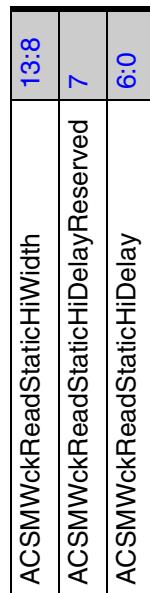


**Table 13-429 Fields for Register: ACSMWckReadStaticLoPulse\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
13:8	ACSMWckReadStaticLoWidth	R/W	ACSMWckReadStaticLoWidth: The number of MEMCLKs hwt_wck_en is asserted <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	ACSMWckReadStaticLoDelayRese rved	R/W	ACSMWckReadStaticLoDelayReserved: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6:0	ACSMWckReadStaticLoDelay	R/W	ACSMWckReadStaticLoDelay: The number of MEMCLKs between the cycle that the WCKRDSLO bit is set in the ACSM and the cycle that hwt_wck_en is asserted. Max=95 <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.94 ACSMWckReadStaticHiPulse\_pX (for X = 0; X <= 3)

- **Name:** Timing Parameters for WckReadStaticHi Pulse
- **Description:** ACSMWckReadStaticHiPulse\_pX: Contains the delay and pulse width for WckReadStaticHi
- **Size:** 14 bits
- **Offset:** 0x70000+0x13a+(X\*0x100000)
- **Exists:** Always



**Table 13-430 Fields for Register: ACSMWckReadStaticHiPulse\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
13:8	ACSMWckReadStaticHiWidth	R/W	ACSMWckReadStaticHiWidth: The number of MEMCLKs hwt_wck_en and hwt_wck_toggle are asserted <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	ACSMWckReadStaticHiDelayRese rved	R/W	ACSMWckReadStaticHiDelayReserved: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6:0	ACSMWckReadStaticHiDelay	R/W	ACSMWckReadStaticHiDelay: The number of MEMCLKs between the cycle that the WCKRDSHI bit is set in the ACSM and the cycle that hwt_wck_en and hwt_wck_toggle are asserted. hwt_wck_toggle is asserted to 'b01. Max=95 <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.95 ACSMWckReadTogglePulse\_pX (for X = 0; X <= 3)

- **Name:** Timing Parameters for WckReadToggle Pulse
- **Description:** ACSMWckReadTogglePulse\_pX: Contains the delay and pulse width for WckReadToggle
- **Size:** 14 bits
- **Offset:** 0x70000+0x13b+(X\*0x100000)
- **Exists:** Always

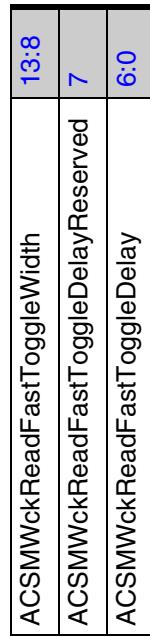


**Table 13-431 Fields for Register: ACSMWckReadTogglePulse\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
13:8	ACSMWckReadToggleWidth	R/W	ACSMWckReadToggleWidth: The number of MEMCLKs hwt_wck_en and hwt_wck_toggle are asserted <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	ACSMWckReadToggleDelayReserved	R/W	ACSMWckReadToggleDelayReserved: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6:0	ACSMWckReadToggleDelay	R/W	ACSMWckReadToggleDelay: The number of MEMCLKs between the cycle that the WCKRDRTG bit is set in the ACSM and the cycle that hwt_wck_en and hwt_wck_toggle are asserted. hwt_wck_toggle is asserted to 'b10. Max=95 <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.96 ACSMWckReadFastTogglePulse\_pX (for X = 0; X <= 3)

- **Name:** Timing Parameters for WckReadFastToggle Pulse
- **Description:** ACSMWckReadFastTogglePulse\_pX: Contains the delay and pulse width for WckReadFastToggle
- **Size:** 14 bits
- **Offset:** 0x70000+0x13c+(X\*0x100000)
- **Exists:** Always



**Table 13-432 Fields for Register: ACSMWckReadFastTogglePulse\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
13:8	ACSMWckReadFastToggleWidth	R/W	ACSMWckReadFastToggleWidth: The number of MEMCLKs hwt_wck_en and hwt_wck_toggle are asserted <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	ACSMWckReadFastToggleDelayReserved	R/W	ACSMWckReadFastToggleDelayReserved: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6:0	ACSMWckReadFastToggleDelay	R/W	ACSMWckReadFastToggleDelay: The number of MEMCLKs between the cycle that the WCKRDRFTG bit is set in the ACSM and the cycle that hwt_wck_en and hwt_wck_toggle are asserted. hwt_wck_toggle is asserted to 'b11. Max=95 <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.97 ACSMWckFreqSwStaticLoPulse\_pX (for X = 0; X <= 3)

- **Name:** Timing Parameters for WckFreqSwStaticLo Pulse
- **Description:** ACSMWckFreqSwStaticLoPulse\_pX: Contains the delay and pulse width for WckFreqSwStaticLo
- **Size:** 14 bits
- **Offset:** 0x70000+0x13d+(X\*0x100000)
- **Exists:** Always

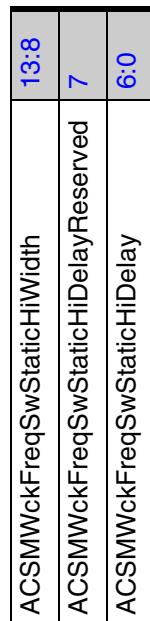


**Table 13-433 Fields for Register: ACSMWckFreqSwStaticLoPulse\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
13:8	ACSMWckFreqSwStaticLoWidth	R/W	ACSMWckFreqSwStaticLoWidth: The number of MEMCLKs hwt_wck_en is asserted <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	ACSMWckFreqSwStaticLoDelayReserved	R/W	ACSMWckFreqSwStaticLoDelayReserved: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6:0	ACSMWckFreqSwStaticLoDelay	R/W	ACSMWckFreqSwStaticLoDelay: The number of MEMCLKs between the cycle that the WCKFSSL0 bit is set in the ACSM and the cycle that hwt_wck_en is asserted. Max=95 <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.98 ACSMWckFreqSwStaticHiPulse\_pX (for X = 0; X <= 3)

- **Name:** Timing Parameters for WckFreqSwStaticHi Pulse
- **Description:** ACSMWckFreqSwStaticHiPulse\_pX: Contains the delay and pulse width for WckFreqSwStaticHi
- **Size:** 14 bits
- **Offset:** 0x70000+0x13e+(X\*0x100000)
- **Exists:** Always



**Table 13-434 Fields for Register: ACSMWckFreqSwStaticHiPulse\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
13:8	ACSMWckFreqSwStaticHiWidth	R/W	ACSMWckFreqSwStaticHiWidth: The number of MEMCLKs hwt_wck_en and hwt_wck_toggle are asserted <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	ACSMWckFreqSwStaticHiDelayReserved	R/W	ACSMWckFreqSwStaticHiDelayReserved: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6:0	ACSMWckFreqSwStaticHiDelay	R/W	ACSMWckFreqSwStaticHiDelay: The number of MEMCLKs between the cycle that the WCKFSSH1 bit is set in the ACSM and the cycle that hwt_wck_en and hwt_wck_toggle are asserted. hwt_wck_toggle is asserted to 'b01. Max=95 <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.99 ACSMWckFreqSwTogglePulse\_pX (for X = 0; X <= 3)

- **Name:** Timing Parameters for WckFreqSwToggle Pulse
- **Description:** ACSMWckFreqSwTogglePulse\_pX: Contains the delay and pulse width for WckFreqSwToggle
- **Size:** 14 bits
- **Offset:** 0x70000+0x13f+(X\*0x100000)
- **Exists:** Always



**Table 13-435 Fields for Register: ACSMWckFreqSwTogglePulse\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
13:8	ACSMWckFreqSwToggleWidth	R/W	ACSMWckFreqSwToggleWidth: The number of MEMCLKs hwt_wck_en and hwt_wck_toggle are asserted <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	ACSMWckFreqSwToggleDelayReserved	R/W	ACSMWckFreqSwToggleDelayReserved: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6:0	ACSMWckFreqSwToggleDelay	R/W	ACSMWckFreqSwToggleDelay: The number of MEMCLKs between the cycle that the WCKFSRTG bit is set in the ACSM and the cycle that hwt_wck_en and hwt_wck_toggle are asserted. hwt_wck_toggle is asserted to 'b10. Max=95 <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.100 ACSMWckFreqSwFastTogglePulse\_pX (for X = 0; X <= 3)

- **Name:** Timing Parameters for WckFreqSwFastToggle Pulse
- **Description:** ACSMWckFreqSwFastTogglePulse\_pX: Contains the delay and pulse width for WckFreqSwFastToggle
- **Size:** 14 bits
- **Offset:** 0x70000+0x140+(X\*0x100000)
- **Exists:** Always



**Table 13-436 Fields for Register: ACSMWckFreqSwFastTogglePulse\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
13:8	ACSMWckFreqSwFastToggleWidth	R/W	ACSMWckFreqSwFastToggleWidth: The number of MEMCLKs hwt_wck_en and hwt_wck_toggle are asserted <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	ACSMWckFreqSwFastToggleDelay Reserved	R/W	ACSMWckFreqSwFastToggleDelayReserved: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

Bits	Name	Memory Access	Description
6:0	ACSMWckFreqSwFastToggleDelay	R/W	<p>ACSMWckFreqSwFastToggleDelay: The number of MEMCLKs between the cycle that the WCKFSRFTG bit is set in the ACSM and the cycle that hwt_wck_en and hwt_wck_toggle are asserted. hwt_wck_toggle is asserted to 'b11. Max=95</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.101 ACSMWckFreeRunMode\_pX (for X = 0; X <= 3)

- **Description:** ACSMWckFreeRunMode\_pX: Extends wck pulses in free running mode
- **Size:** 1 bit
- **Offset:** 0x70000+0x141+(X\*0x100000)
- **Exists:** Always



**Table 13-437 Fields for Register: ACSMWckFreeRunMode\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
0	ACSMWckFreeRunMode_p0	R/W	<p>ACSMWckFreeRunMode_p0: If csrACSMWckFreeRunMode = 1 AND csrACSMWckEnControl = 0x01 AND WCKTG=1,then</p> <ul style="list-style-type: none"> <li>■ 1: next time wck_en is set, keep it set</li> <li>■ 2: next time wck_toggle goes to 3, keep it at 3.</li> <li>■ 3: wck_cs is held at current value</li> </ul> <p>The next time WCKTG=1, the override on wck_en, wck_cs, and wck_toggle is disengaged</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.102 ACSMLowSpeedClockEnable

- **Description:** ACSMLowSpeedClockEnable: Used to control and reduce speed of ACSM Output and AC bump clocks
- **Size:** 10 bits
- **Offset:** 0x70000+0x142
- **Exists:** Always



**Table 13-438 Fields for Register: ACSMLowSpeedClockEnable**

Bits	Name	Memory Access	Description
9:0	ACSMLowSpeedClockEnable	R/W	<p>ACSMLowSpeedClockEnable: The low-speed clock feature is enabled when bits 6:0 are non-zero.</p> <ul style="list-style-type: none"> <li>■ When bit 7 is 0, the clock to the ACSM logic is divided by <math>2^{*}\text{csrACSMLowSpeedClockEnable}[6:0]</math></li> <li>■ When bit 7 is 1, the ACSM runs at full speed.</li> <li>■ When bits 9:8 = 0, the AC bump clock is divided by <math>2^{*}\text{csrACSMLowSpeedClockEnable}[6:0]</math></li> <li>■ When bits 9:8 = 1, the AC bump clock is divided by <math>4^{*}\text{csrACSMLowSpeedClockEnable}[6:0]</math></li> <li>■ When bits 9:8 = 2, the AC bump clock is divided by <math>1^{*}\text{csrACSMLowSpeedClockEnable}[6:0]</math></li> </ul> <p><b>Note:</b> When bits 9:8 = 2, <math>\text{csrACSMLowSpeedClockEnable}[6:0]</math> must be an even value <math>\geq 2</math>.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.103 ACSMLowSpeedClockDelay

- **Description:** ACSMLowSpeedClockDelay: Used to offset reduced speed AC bump clocks
- **Size:** 9 bits
- **Offset:** 0x70000+0x144
- **Exists:** Always



**Table 13-439 Fields for Register: ACSMLowSpeedClockDelay**

Bits	Name	Memory Access	Description
8:0	ACSMLowSpeedClockDelay	R/W	<p>ACSMLowSpeedClockDelay: Used to specify an offset between the beginning of the ACSM sequence and the first clock edge. Unit is DfiClks</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.104 ACSMRptCntOverride\_pX (for X = 0; X <= 3)

- **Description:** ACSMRptCntOverride\_pX: Used to override the REPEATCNT field of the ACSM Control Word
- **Size:** 8 bits
- **Offset:** 0x70000+0x145+(X\*0x100000)
- **Exists:** Always



**Table 13-440 Fields for Register: ACSMRptCntOverride\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
7:0	ACSMRptCntOverride_p0	R/W	<p>ACSMRptCntOverride_p0: When ACSMRptCntOverrideEn is 1, and the RCNE0 field of the ACSM Control Word is 1, REPEATCNT is taken from this register. It is subject to doubling if csrACSMRptCntDbl=1; When ACSMRptCntOverrideEn is 1, this register must be non-zero.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.105 ACSMRptCntDbl\_pX (for X = 0; X <= 3)

- **Description:** ACSMRptCntDbl\_pX: Used to double the REPEATCNT field of the ACSM Control Word
- **Size:** 1 bit
- **Offset:** 0x70000+0x146+(X\*0x100000)
- **Exists:** Always



**Table 13-441 Fields for Register: ACSMRptCntDbl\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
0	ACSMRptCntDbl_p0	R/W	ACSMRptCntDbl_p0: When set, the REPEATCNT is doubled from the value in the ACSM Control Word <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.106 ACSMParityStatus

- **Description:** ACSMParityStatus: ACSM Byte Parity Error Status
- **Size:** 8 bits
- **Offset:** 0x70000+0x147
- **Exists:** Always



**Table 13-442 Fields for Register: ACSMParityStatus**

Bits	Name	Memory Access	Description
7:0	ACSMParityStatus	R	<p>ACSMParityStatus: the ACSM supports byte parity on the external memory. If a parity error occurs, this register captures which byte(s) are affected by the first error that occurs. The register is cleared when csrACSMRun transitions from 0 -&gt; 1.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.4.107 HwtLpCsEnBypass

- **Description:** HwtLpCsEnBypass: Reserved for PHY training firmware use.
- **Size:** 1 bit
- **Offset:** 0x70000+0x174
- **Exists:** Always



Table 13-443 Fields for Register: HwtLpCsEnBypass

Bits	Name	Memory Access	Description
0	HwtLpCsEnBypass	R/W	<p>HwtLpCsEnBypass: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.108 ACSMNopAddr

- **Description:** ACSMNopAddr: Enable ACSMXlat address dereferencing
- **Size:** 7 bits
- **Offset:** 0x70000+0x18a
- **Exists:** Always



**Table 13-444 Fields for Register: ACSMNopAddr**

Bits	Name	Memory Access	Description
6:0	ACSMNopAddr	R/W	<p>ACSMNopAddr: When Enabled ACSMStartAddr[10:1] and ACSMStopAddr[10:1] address are sourced from ACSMPtrXlat table instead of directly using the registers. The ACSMStartAddr[0], is always driven 0 when AcsmPtrXlat is used. Address of an ACSM nop instruction pair, used for AcsmNopFlag feature.</p> <ul style="list-style-type: none"> <li>■ bits [9:8] of the Nop Addr are fixed 0's. bit [0] is forced 0/1 based on start/stop address.</li> <li>■ bits [7:1] are sourced from this field.</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.109 SnoopCntrl

- **Description:** SnoopCntrl: Selects the value to be driven on hwt\_snoop\_en in HWT mode.
- **Size:** 4 bits
- **Offset:** 0x70000+0x1a7
- **Exists:** Always



**Table 13-445 Fields for Register: SnoopCntrl**

Bits	Name	Memory Access	Description
3:0	SnoopCntrl	R/W	<p>SnoopCntrl: This csr needs to set to 0 by default. hwt_snoop_en will be driven with a value as programmed in csr. Example Usage in LPDDR4X Mode: Set this csr SnoopCntrl = 0x1 (MR18, LSB) Program ACSM to issue MR18 Read Command. Set this csr SnoopCntrl = 0x2 (MR19, MSB) Program ACSM to issue MR19 Read Command. PUB Dbyte Module will compute the new TxDqDly value based on the Osc value read from above two read commands. Program ForceInternalUpdate or PulseDbyteDIIUpdatePhase for the new Delay value to take effect.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.110 ACSMParityInvert

- **Description:** ACSMParityInvert: Forces parity errors in ACSM RAM
- **Size:** 8 bits
- **Offset:** 0x70000+0x1a8
- **Exists:** Always



**Table 13-446 Fields for Register: ACSMParityInvert**

Bits	Name	Memory Access	Description
7:0	ACSMParityInvert	R/W	ACSMParityInvert: When set, the corresponding parity bit is inverted on ACSM RAM writes <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.111 AcsmPsIdx

- **Description:** AcsmPsIdx: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 2 bits
- **Offset:** 0x70000+0x1a9
- **Exists:** Always



**Table 13-447 Fields for Register: AcsmPsIdx**

Bits	Name	Memory Access	Description
1:0	AcsmPsIdx	R/W	<p>AcsmPsIdx: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.112 AcsmDynPtrCtrl

- **Description:** AcsmDynPtrCtrl: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 3 bits
- **Offset:** 0x70000+0x1aa
- **Exists:** Always

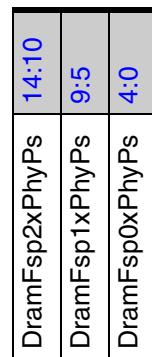


**Table 13-448 Fields for Register: AcsmDynPtrCtrl**

Bits	Name	Memory Access	Description
2:0	AcsmDynPtrCtrl	R/W	AcsmDynPtrCtrl: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.113 FspState

- **Name:** Captures the state of status interface for retention enter/exit
- **Description:** FspState: Captures the state of DFI status interface before entering retention. Each field corresponds to the PHY PState loaded in to in DRAM FSP. Value of 0xf indicates DRAM FSP is unloaded and can be considered invalid.
- **Size:** 15 bits
- **Offset:** 0x70000+0x1ef
- **Exists:** Always

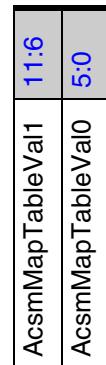


**Table 13-449 Fields for Register: FspState**

Bits	Name	Memory Access	Description
14:10	DramFsp2xPhyPs	R/W	DramFsp2xPhyPs: PHY PState loaded in FSP2. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
9:5	DramFsp1xPhyPs	R/W	DramFsp1xPhyPs: PHY PState loaded in FSP1. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
4:0	DramFsp0xPhyPs	R/W	DramFsp0xPhyPs: PHY PState loaded in FSP0. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.114 AcsmMapTableX (for X = 0; X <= 63)

- **Description:** AcsmMapTableX: Maps ProgPtr+DynPtr into XlatTable Address
- **Size:** 12 bits
- **Offset:** 0x70000+0x200+(X\*0x1)
- **Exists:** Always



**Table 13-450 Fields for Register: AcsmMapTableX (for X = 0; X <= 63)**

Bits	Name	Memory Access	Description
11:6	AcsmMapTableVal1	R/W	AcsmMapTableVal1: Mapping for [ProgPtr,DynPtr] = 1 <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
5:0	AcsmMapTableVal0	R/W	AcsmMapTableVal0: Mapping for [ProgPtr,DynPtr] = 0 <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.4.115 AcsmStartAddrXlatValX (for X = 0; X <= 63)

- **Description:** AcsmStartAddrXlatValX: Start Address value of AcsmPtr Translation Register0
- **Size:** 10 bits
- **Offset:** 0x70000+0x324+(X\*0x1)
- **Exists:** Always



**Table 13-451 Fields for Register: AcsmStartAddrXlatValX (for X = 0; X <= 63)**

Bits	Name	Memory Access	Description
9:0	AcsmStartAddrXlatVal0	R/W	<p>AcsmStartAddrXlatVal0: Start Address value of AcsmPtr Translation Register0 Programmed by dwc_ddrphy_phyinit_C_initPhyConfig() to support frequency changes. ACSM Start address bits 9:0. Start address[0] is forced to 0.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.4.116 AcsmStopAddrXlatValX (for X = 0; X <= 63)

- **Description:** AcsmStopAddrXlatValX: Stop Address value of AcsmPtr Translation Register0
- **Size:** 10 bits
- **Offset:** 0x70000+0x38b+(X\*0x1)
- **Exists:** Always



**Table 13-452 Fields for Register: AcsmStopAddrXlatValX (for X = 0; X <= 63)**

Bits	Name	Memory Access	Description
9:0	AcsmStopAddrXlatVal0	R/W	<p>AcsmStopAddrXlatVal0: Stop Address value of AcsmPtr Translation Register0 Programmed by dwc_ddrphy_phyinit_C_initPhyConfig() to support frequency changes. ACSM Stop address bits 9:0. Stop address[0] is forced to 1.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

## 13.5 **DWC\_DDRPHYA\_INITENGj\_Pk Registers**

### 13.5.1 DVFSCEn\_pX (for X = 0; X <= 3)

- **Description:** DVFSCEn\_pX: Deprecate
- **Size:** 1 bit
- **Offset:** 0x90000+0x10+(X\*0x100000)
- **Exists:** Always



**Table 13-453 Fields for Register: DVFSCEn\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
0	DVFSCEn_p0	R/W	<p>DVFSCEn_p0: Deprecate</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.2 MtestMuxSel

- **Description:** MtestMuxSel: Digital Observation Pin control
- **Size:** 10 bits
- **Offset:** 0x90000+0x1a
- **Exists:** Always



**Table 13-454 Fields for Register: MtestMuxSel**

Bits	Name	Memory Access	Description
9:0	MtestMuxSel	R/W	<p>MtestMuxSel: Controls for the mux for asynchronous data to the Digital Observation Pin.</p> <ul style="list-style-type: none"> <li>■ Encoding 9'h0 causes this chiplet to drive 0, (allowing flat 'OR' of pass-through information).</li> <li>■ MtestMuxSel[4:0] - Lower 5 bits selects one bit from the 32 bit MtestMux in each section or slice.</li> </ul> <p>Detailed tables are in the PUB Databook for PUB sections (AC, MASTER, etc.)</p> <ul style="list-style-type: none"> <li>■ MtestMuxSel[8:5] - Where more than one MtestMux exists, non-zero values select the outputs of the additional Mtest-Muxes,           <ul style="list-style-type: none"> <li>□ DBYTE MtestMuxSel[6:5]=2'h0 for Mux-A and 2'h1 for Mux-B and 2'h2 for Mux-C.</li> <li>□ For all other slaves, MtestMuxSel[8:5] are unused</li> </ul> </li> </ul> <p><b>Note:</b> See the PUB Databook for how, or if, the Digital Observation Pin is mapped to a physical bump in this configuration.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.3 StartVector0bX (for X = 0; X <= 15)

- **Description:** StartVector0bX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 12 bits
- **Offset:** 0x90000+0x1c+(X\*0x1)
- **Exists:** Always

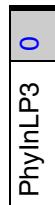


**Table 13-455 Fields for Register: StartVector0bX (for X = 0; X <= 15)**

Bits	Name	Memory Access	Description
11:0	StartVector0b0	R/W	StartVector0b0: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.5.4 PhyInLPX (for X = 0; X <= 3)

- **Description:** PhyInLPX: Indicator for PIE Lower Power 3 (LP3) Status
- **Size:** 1 bit
- **Offset:** 0x90000+0x2d
- **Exists:** Always

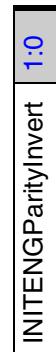


**Table 13-456 Fields for Register: PhyInLPX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
0	PhyInLP3	R/W	<p>PhyInLP3: Read Only. Set to 1 by the PIE once completed LP3 Entry sequence; Cleared during LP3 Exit sequence. System software can read this csr for the status of PIE engine.</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

### 13.5.5 INITENGParityInvert

- **Description:** INITENGParityInvert: Invert APB Parity for register slave INITENG
- **Size:** 2 bits
- **Offset:** 0x90000+0x4d
- **Exists:** Always



**Table 13-457 Fields for Register: INITENGParityInvert**

Bits	Name	Memory Access	Description
1:0	INITENGParityInvert	R/W	<p>INITENGParityInvert: Invert APB Parity for register slave INITENG. As required for Automotive. NOTE: This register should be used only for test. Set the bits for only one slave at a time. When bits are set for a particular slave. APB Reads of only that slave are valid. Bit 0 applies to [7:0] Bit 1 applies to [15:8] In HMZCAL, When ZcalStopClk=1, this csr can not be accessed.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.6 ScratchPadINITENG

- **Description:** ScratchPadINITENG: ScratchPad for INITENG
- **Size:** 16 bits
- **Offset:** 0x90000+0x7d
- **Exists:** Always



**Table 13-458 Fields for Register: ScratchPadINITENG**

Bits	Name	Memory Access	Description
15:0	ScratchPadINITENG	R/W	<p>ScratchPadINITENG: ScratchPad for INITENG. As required for Automotive. In HMZCAL, When ZcalStopClk=1, this csr can not be accessed.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.7 PieCtrlStartVec0\_pX (for X = 0; X <= 3)

- **Description:** PieCtrlStartVec0\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 12 bits
- **Offset:** 0x90000+0x708+(X\*0x100000)
- **Exists:** Always

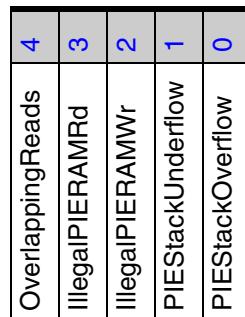


Table 13-459 Fields for Register: PieCtrlStartVec0\_pX (for X = 0; X <= 3)

Bits	Name	Memory Access	Description
11:0	PieCtrlStartVec0_p0	R/W	<p>PieCtrlStartVec0_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.8 PIEProgErrStatus

- **Name:** PIE Programming Error Status
- **Description:** PIEProgErrStatus: If a PIE Programming error occurs, this register says which type of error occurred. The register is cleared each time the PIE starts.
- **Size:** 5 bits
- **Offset:** 0x90000+0x709
- **Exists:** Always



**Table 13-460 Fields for Register: PIEProgErrStatus**

Bits	Name	Memory Access	Description
4	OverlappingReads	R	<p>OverlappingReads: PIE attempted to execute a READ while an earlier one had not completed</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
3	IllegalPIERAMRd	R	<p>IllegalPIERAMRd: There was a CfgBus Rd to PIE RAM while the PIE was running</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
2	IllegalPIERAMWr	R	<p>IllegalPIERAMWr: There was a write to PIE RAM while the PIE was running</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

Bits	Name	Memory Access	Description
1	PIEStackUnderflow	R	<p>PIEStackUnderflow: A RET instruction without a CALL</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
0	PIEStackOverflow	R	<p>PIEStackOverflow: Too many CALL instructions without a RET</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.5.9 PIEParityInvert

- **Description:** PIEParityInvert: Forces parity errors in PIE RAM
- **Size:** 6 bits
- **Offset:** 0x90000+0x70a
- **Exists:** Always



**Table 13-461 Fields for Register: PIEParityInvert**

Bits	Name	Memory Access	Description
5:0	PIEParityInvert	R/W	PIEParityInvert: When set, the corresponding parity bit is inverted on PIE RAM writes <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.5.10 PIEParityStatus

- **Description:** PIEParityStatus: PIE Byte Parity Error Status
- **Size:** 6 bits
- **Offset:** 0x90000+0x70b
- **Exists:** Always



**Table 13-462 Fields for Register: PIEParityStatus**

Bits	Name	Memory Access	Description
5:0	PIEParityStatus	R	<p>PIEParityStatus: If a PIE RAM parity error occurs, this register captures which parity bit was the first to detect an error. The register is cleared each time the PIE starts.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.5.11 Seq0BDisableFlagX (for X = 0; X <= 31)

- **Description:** Seq0BDisableFlagX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x70c+(X\*0x1)
- **Exists:** Always



**Table 13-463 Fields for Register: Seq0BDisableFlagX (for X = 0; X <= 31)**

Bits	Name	Memory Access	Description
15:0	Seq0BDisableFlag0	R/W	<p>Seq0BDisableFlag0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.12 Seq0BCSRUpperWrData

- **Description:** Seq0BCSRUpperWrData: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x71c
- **Exists:** Always



**Table 13-464 Fields for Register: Seq0BCSRUpperWrData**

Bits	Name	Memory Access	Description
15:0	Seq0BCSRUpperWrData	R/W	<p>Seq0BCSRUpperWrData: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.13 Seq0BCSRUpperRdData

- **Description:** Seq0BCSRUpperRdData: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x71d
- **Exists:** Always



**Table 13-465 Fields for Register: Seq0BCSRUpperRdData**

Bits	Name	Memory Access	Description
15:0	Seq0BCSRUpperRdData	R/W	<p>Seq0BCSRUpperRdData: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.14 WaitCondUC

- **Description:** WaitCondUC: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 1 bit
- **Offset:** 0x90000+0x71e
- **Exists:** Always



**Table 13-466 Fields for Register: WaitCondUC**

Bits	Name	Memory Access	Description
0	WaitCondUC	R/W	<p>WaitCondUC: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.15 PhyMstrPMValOverride

- **Description:** PhyMstrPMValOverride: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 4 bits
- **Offset:** 0x90000+0x71f
- **Exists:** Always



**Table 13-467 Fields for Register: PhyMstrPMValOverride**

Bits	Name	Memory Access	Description
3:0	PhyMstrPMValOverride	R/W	<p>PhyMstrPMValOverride: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.16 PIEDebugStatus

- **Description:** PIEDebugStatus: Assorted status signals to help with PIE debug
- **Size:** 4 bits
- **Offset:** 0x90000+0x720
- **Exists:** Always

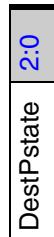


**Table 13-468 Fields for Register: PIEDebugStatus**

Bits	Name	Memory Access	Description
3:0	PIEDebugStatus	R	PIEDebugStatus: Assorted status signals to help with PIE debug <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always <b>Reset Mask:</b> 0x0 <b>Volatile:</b> true

### 13.5.17 DestPstate

- **Description:** DestPstate: Read by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 3 bits
- **Offset:** 0x90000+0x721
- **Exists:** Always



**Table 13-469 Fields for Register: DestPstate**

Bits	Name	Memory Access	Description
2:0	DestPstate	R	<p>DestPstate: Read by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.5.18 DfiFreqXlatDestPStateX (for X = 0; X <= 15)

- **Name:** DFI Frequency Translation Register 0 for Dest PState
- **Description:** DfiFreqXlatDestPStateX: DFI Frequency Translation Register 0 for Dest PState  
Programmed by dwc\_ddrphy\_phyinit\_C\_initPhyConfig() to support frequency changes.
- **Size:** 12 bits
- **Offset:** 0x90000+0x730+(X\*0x1)
- **Exists:** Always

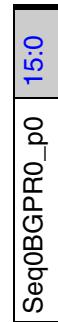


**Table 13-470 Fields for Register: DfiFreqXlatDestPStateX (for X = 0; X <= 15)**

Bits	Name	Memory Access	Description
11:9	DfiFreqXlatDestPStateVal3	R/W	<p>DfiFreqXlatDestPStateVal3: The Dest PState used when dfi_freq value is 3.  <b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
8:6	DfiFreqXlatDestPStateVal2	R/W	<p>DfiFreqXlatDestPStateVal2: The Dest PState used when dfi_freq value is 2.  <b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
5:3	DfiFreqXlatDestPStateVal1	R/W	<p>DfiFreqXlatDestPStateVal1: The Dest PState used when dfi_freq value is 1.  <b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
2:0	DfiFreqXlatDestPStateVal0	R/W	<p>DfiFreqXlatDestPStateVal0: The Dest PState used when dfi_freq value is 0.  <b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.5.19 Seq0BGPR0\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR0\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x800+(X\*0x100000)
- **Exists:** Always

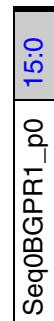


**Table 13-471 Fields for Register: Seq0BGPR0\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR0_p0	R/W	<p>Seq0BGPR0_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.20 Seq0BGPR1\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR1\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x801+(X\*0x100000)
- **Exists:** Always

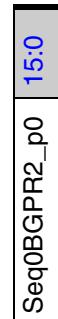


**Table 13-472 Fields for Register: Seq0BGPR1\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR1_p0	R/W	<p>Seq0BGPR1_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.21 Seq0BGPR2\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR2\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x802+(X\*0x100000)
- **Exists:** Always



**Table 13-473 Fields for Register: Seq0BGPR2\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR2_p0	R/W	<p>Seq0BGPR2_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.22 Seq0BGPR3\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR3\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x803+(X\*0x100000)
- **Exists:** Always



**Table 13-474 Fields for Register: Seq0BGPR3\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR3_p0	R/W	<p>Seq0BGPR3_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.23 Seq0BGPR4\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR4\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x804+(X\*0x100000)
- **Exists:** Always

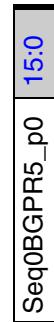


**Table 13-475 Fields for Register: Seq0BGPR4\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR4_p0	R/W	<p>Seq0BGPR4_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.24 Seq0BGPR5\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR5\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x805+(X\*0x100000)
- **Exists:** Always



**Table 13-476 Fields for Register: Seq0BGPR5\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR5_p0	R/W	<p>Seq0BGPR5_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.25 Seq0BGPR6\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR6\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x806+(X\*0x100000)
- **Exists:** Always

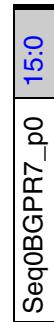


**Table 13-477 Fields for Register: Seq0BGPR6\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR6_p0	R/W	<p>Seq0BGPR6_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.26 Seq0BGPR7\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR7\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x807+(X\*0x100000)
- **Exists:** Always

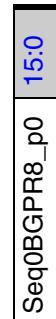


**Table 13-478 Fields for Register: Seq0BGPR7\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR7_p0	R/W	<p>Seq0BGPR7_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.27 Seq0BGPR8\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR8\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x808+(X\*0x100000)
- **Exists:** Always



**Table 13-479 Fields for Register: Seq0BGPR8\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR8_p0	R/W	<p>Seq0BGPR8_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.28 Seq0BGPR9\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR9\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x809+(X\*0x100000)
- **Exists:** Always



**Table 13-480 Fields for Register: Seq0BGPR9\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR9_p0	R/W	<p>Seq0BGPR9_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.29 Seq0BGPR10\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR10\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x80a+(X\*0x100000)
- **Exists:** Always



**Table 13-481 Fields for Register: Seq0BGPR10\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR10_p0	R/W	<p>Seq0BGPR10_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.30 Seq0BGPR11\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR11\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x80b+(X\*0x100000)
- **Exists:** Always

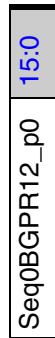


**Table 13-482 Fields for Register: Seq0BGPR11\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR11_p0	R/W	<p>Seq0BGPR11_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.31 Seq0BGPR12\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR12\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x80c+(X\*0x100000)
- **Exists:** Always

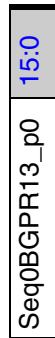


**Table 13-483 Fields for Register: Seq0BGPR12\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR12_p0	R/W	<p>Seq0BGPR12_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.32 Seq0BGPR13\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR13\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x80d+(X\*0x100000)
- **Exists:** Always

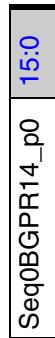


**Table 13-484 Fields for Register: Seq0BGPR13\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR13_p0	R/W	<p>Seq0BGPR13_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.33 Seq0BGPR14\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR14\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x80e+(X\*0x100000)
- **Exists:** Always

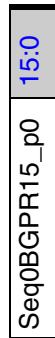


**Table 13-485 Fields for Register: Seq0BGPR14\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR14_p0	R/W	<p>Seq0BGPR14_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.34 Seq0BGPR15\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR15\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x80f+(X\*0x100000)
- **Exists:** Always



**Table 13-486 Fields for Register: Seq0BGPR15\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR15_p0	R/W	<p>Seq0BGPR15_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.35 Seq0BGPR16\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR16\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x810+(X\*0x100000)
- **Exists:** Always

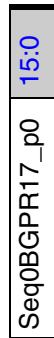


**Table 13-487 Fields for Register: Seq0BGPR16\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR16_p0	R/W	<p>Seq0BGPR16_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.36 Seq0BGPR17\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR17\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x811+(X\*0x100000)
- **Exists:** Always

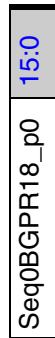


**Table 13-488 Fields for Register: Seq0BGPR17\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR17_p0	R/W	<p>Seq0BGPR17_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.37 Seq0BGPR18\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR18\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x812+(X\*0x100000)
- **Exists:** Always

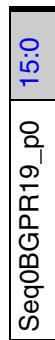


**Table 13-489 Fields for Register: Seq0BGPR18\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR18_p0	R/W	<p>Seq0BGPR18_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.38 Seq0BGPR19\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR19\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x813+(X\*0x100000)
- **Exists:** Always



**Table 13-490 Fields for Register: Seq0BGPR19\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR19_p0	R/W	<p>Seq0BGPR19_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.39 Seq0BGPR20\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR20\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x814+(X\*0x100000)
- **Exists:** Always

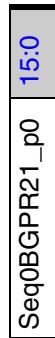


**Table 13-491 Fields for Register: Seq0BGPR20\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR20_p0	R/W	<p>Seq0BGPR20_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.40 Seq0BGPR21\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR21\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x815+(X\*0x100000)
- **Exists:** Always

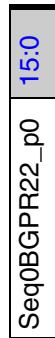


**Table 13-492 Fields for Register: Seq0BGPR21\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR21_p0	R/W	<p>Seq0BGPR21_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.41 Seq0BGPR22\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR22\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x816+(X\*0x100000)
- **Exists:** Always

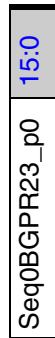


**Table 13-493 Fields for Register: Seq0BGPR22\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR22_p0	R/W	<p>Seq0BGPR22_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.42 Seq0BGPR23\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR23\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x817+(X\*0x100000)
- **Exists:** Always

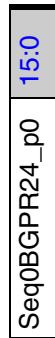


**Table 13-494 Fields for Register: Seq0BGPR23\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR23_p0	R/W	<p>Seq0BGPR23_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.43 Seq0BGPR24\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR24\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x818+(X\*0x100000)
- **Exists:** Always



**Table 13-495 Fields for Register: Seq0BGPR24\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR24_p0	R/W	<p>Seq0BGPR24_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.44 Seq0BGPR25\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR25\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x819+(X\*0x100000)
- **Exists:** Always



**Table 13-496 Fields for Register: Seq0BGPR25\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR25_p0	R/W	<p>Seq0BGPR25_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.45 Seq0BGPR26\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR26\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x81a+(X\*0x100000)
- **Exists:** Always

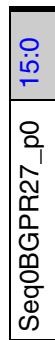


**Table 13-497 Fields for Register: Seq0BGPR26\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR26_p0	R/W	<p>Seq0BGPR26_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.46 Seq0BGPR27\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR27\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x81b+(X\*0x100000)
- **Exists:** Always



**Table 13-498 Fields for Register: Seq0BGPR27\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR27_p0	R/W	<p>Seq0BGPR27_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.47 Seq0BGPR28\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR28\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x81c+(X\*0x100000)
- **Exists:** Always

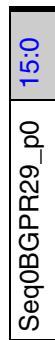


**Table 13-499 Fields for Register: Seq0BGPR28\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR28_p0	R/W	<p>Seq0BGPR28_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.48 Seq0BGPR29\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR29\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x81d+(X\*0x100000)
- **Exists:** Always



**Table 13-500 Fields for Register: Seq0BGPR29\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR29_p0	R/W	<p>Seq0BGPR29_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.49 Seq0BGPR30\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR30\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x81e+(X\*0x100000)
- **Exists:** Always

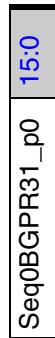


**Table 13-501 Fields for Register: Seq0BGPR30\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR30_p0	R/W	<p>Seq0BGPR30_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.50 Seq0BGPR31\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR31\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x81f+(X\*0x100000)
- **Exists:** Always



**Table 13-502 Fields for Register: Seq0BGPR31\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR31_p0	R/W	<p>Seq0BGPR31_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.51 Seq0BGPRPage

- **Description:** Seq0BGPRPage: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 2 bits
- **Offset:** 0x90000+0x828
- **Exists:** Always



**Table 13-503 Fields for Register: Seq0BGPRPage**

Bits	Name	Memory Access	Description
1:0	Seq0BGPRPage	R/W	<p>Seq0BGPRPage: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.52 Seq0BFixedAddrBits

- **Description:** Seq0BFixedAddrBits: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 9 bits
- **Offset:** 0x90000+0x829
- **Exists:** Always



**Table 13-504 Fields for Register: Seq0BFixedAddrBits**

Bits	Name	Memory Access	Description
8:0	Seq0BFixedAddrBits	R/W	<p>Seq0BFixedAddrBits: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.53 PieDynFlagMaskX (for X = 0; X <= 3)

- **Description:** PieDynFlagMaskX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x82a+(X\*0x1)
- **Exists:** Always



**Table 13-505 Fields for Register: PieDynFlagMaskX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	PieDynFlagMask0	R/W	<p>PieDynFlagMask0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.54 PieDynFlagSel

- **Description:** PieDynFlagSel: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 2 bits
- **Offset:** 0x90000+0x82e
- **Exists:** Always



**Table 13-506 Fields for Register: PieDynFlagSel**

Bits	Name	Memory Access	Description
1:0	PieDynFlagSel	R/W	<p>PieDynFlagSel: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.55 Seq0BPieGPRSel

- **Description:** Seq0BPieGPRSel: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 1 bit
- **Offset:** 0x90000+0x82f
- **Exists:** Always



**Table 13-507 Fields for Register: Seq0BPieGPRSel**

Bits	Name	Memory Access	Description
0	Seq0BPieGPRSel	R/W	<p>Seq0BPieGPRSel: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.56 Seq0BDLY0\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY0\_pX: PHY Initialization Engine (PIE) Delay Register 0
- **Size:** 16 bits
- **Offset:** 0x90000+0x8e0+(X\*0x100000)
- **Exists:** Always



**Table 13-508 Fields for Register: Seq0BDLY0\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY0_p0	R/W	<p>Seq0BDLY0_p0: PHY Initialization Engine (PIE) Delay Register 0 Programmed by dwc_ddrphy_phyinit_l_loadPIEImage() to support frequency changes. The PIE stalls for n DFICLKs after executing instructions that reference a Delay CSR. n is equal to the value in the referenced CSR times 4.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.57 Seq0BDLY1\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY1\_pX: PHY Initialization Engine (PIE) Delay Register 1
- **Size:** 16 bits
- **Offset:** 0x90000+0x8e1+(X\*0x100000)
- **Exists:** Always



**Table 13-509 Fields for Register: Seq0BDLY1\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY1_p0	R/W	<p>Seq0BDLY1_p0: PHY Initialization Engine (PIE) Delay Register 1 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.58 Seq0BDLY2\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY2\_pX: PHY Initialization Engine (PIE) Delay Register 2
- **Size:** 16 bits
- **Offset:** 0x90000+0x8e2+(X\*0x100000)
- **Exists:** Always

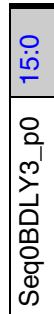


**Table 13-510 Fields for Register: Seq0BDLY2\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY2_p0	R/W	<p>Seq0BDLY2_p0: PHY Initialization Engine (PIE) Delay Register 2 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.59 Seq0BDLY3\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY3\_pX: PHY Initialization Engine (PIE) Delay Register 3
- **Size:** 16 bits
- **Offset:** 0x90000+0x8e3+(X\*0x100000)
- **Exists:** Always



**Table 13-511 Fields for Register: Seq0BDLY3\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY3_p0	R/W	<p>Seq0BDLY3_p0: PHY Initialization Engine (PIE) Delay Register 3 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.60 Seq0BDLY4\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY4\_pX: PHY Initialization Engine (PIE) Delay Register 4
- **Size:** 16 bits
- **Offset:** 0x90000+0x8e4+(X\*0x100000)
- **Exists:** Always



**Table 13-512 Fields for Register: Seq0BDLY4\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY4_p0	R/W	<p>Seq0BDLY4_p0: PHY Initialization Engine (PIE) Delay Register 4 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.61 Seq0BDLY5\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY5\_pX: PHY Initialization Engine (PIE) Delay Register 5
- **Size:** 16 bits
- **Offset:** 0x90000+0x8e5+(X\*0x100000)
- **Exists:** Always



**Table 13-513 Fields for Register: Seq0BDLY5\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY5_p0	R/W	<p>Seq0BDLY5_p0: PHY Initialization Engine (PIE) Delay Register 5 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.62 Seq0BDLY6\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY6\_pX: PHY Initialization Engine (PIE) Delay Register 6
- **Size:** 16 bits
- **Offset:** 0x90000+0x8e6+(X\*0x100000)
- **Exists:** Always



**Table 13-514 Fields for Register: Seq0BDLY6\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY6_p0	R/W	<p>Seq0BDLY6_p0: PHY Initialization Engine (PIE) Delay Register 6 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.63 Seq0BDLY7\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY7\_pX: PHY Initialization Engine (PIE) Delay Register 7
- **Size:** 16 bits
- **Offset:** 0x90000+0x8e7+(X\*0x100000)
- **Exists:** Always



**Table 13-515 Fields for Register: Seq0BDLY7\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY7_p0	R/W	<p>Seq0BDLY7_p0: PHY Initialization Engine (PIE) Delay Register 7 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.64 Seq0BDLY8\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY8\_pX: PHY Initialization Engine (PIE) Delay Register 8
- **Size:** 16 bits
- **Offset:** 0x90000+0x8e8+(X\*0x100000)
- **Exists:** Always



**Table 13-516 Fields for Register: Seq0BDLY8\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY8_p0	R/W	<p>Seq0BDLY8_p0: PHY Initialization Engine (PIE) Delay Register 8 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.65 Seq0BDLY9\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY9\_pX: PHY Initialization Engine (PIE) Delay Register 9
- **Size:** 16 bits
- **Offset:** 0x90000+0x8e9+(X\*0x100000)
- **Exists:** Always



**Table 13-517 Fields for Register: Seq0BDLY9\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY9_p0	R/W	<p>Seq0BDLY9_p0: PHY Initialization Engine (PIE) Delay Register 9 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.66 Seq0BDLY10\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY10\_pX: PHY Initialization Engine (PIE) Delay Register 10
- **Size:** 16 bits
- **Offset:** 0x90000+0x8ea+(X\*0x100000)
- **Exists:** Always

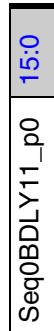


**Table 13-518 Fields for Register: Seq0BDLY10\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY10_p0	R/W	<p>Seq0BDLY10_p0: PHY Initialization Engine (PIE) Delay Register 10 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.67 Seq0BDLY11\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY11\_pX: PHY Initialization Engine (PIE) Delay Register 11
- **Size:** 16 bits
- **Offset:** 0x90000+0x8eb+(X\*0x100000)
- **Exists:** Always

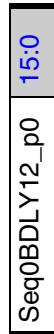


**Table 13-519 Fields for Register: Seq0BDLY11\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY11_p0	R/W	<p>Seq0BDLY11_p0: PHY Initialization Engine (PIE) Delay Register 11 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.68 Seq0BDLY12\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY12\_pX: PHY Initialization Engine (PIE) Delay Register 12
- **Size:** 16 bits
- **Offset:** 0x90000+0x8ec+(X\*0x100000)
- **Exists:** Always

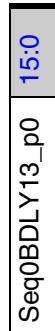


**Table 13-520 Fields for Register: Seq0BDLY12\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY12_p0	R/W	<p>Seq0BDLY12_p0: PHY Initialization Engine (PIE) Delay Register 12 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.69 Seq0BDLY13\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY13\_pX: PHY Initialization Engine (PIE) Delay Register 13
- **Size:** 16 bits
- **Offset:** 0x90000+0x8ed+(X\*0x100000)
- **Exists:** Always

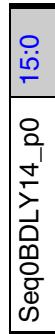


**Table 13-521 Fields for Register: Seq0BDLY13\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY13_p0	R/W	<p>Seq0BDLY13_p0: PHY Initialization Engine (PIE) Delay Register 13 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.70 Seq0BDLY14\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY14\_pX: PHY Initialization Engine (PIE) Delay Register 14
- **Size:** 16 bits
- **Offset:** 0x90000+0x8ee+(X\*0x100000)
- **Exists:** Always

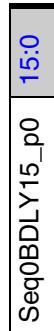


**Table 13-522 Fields for Register: Seq0BDLY14\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY14_p0	R/W	<p>Seq0BDLY14_p0: PHY Initialization Engine (PIE) Delay Register 14 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.71 Seq0BDLY15\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY15\_pX: PHY Initialization Engine (PIE) Delay Register 15
- **Size:** 16 bits
- **Offset:** 0x90000+0x8ef+(X\*0x100000)
- **Exists:** Always

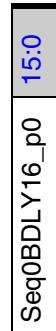


**Table 13-523 Fields for Register: Seq0BDLY15\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY15_p0	R/W	<p>Seq0BDLY15_p0: PHY Initialization Engine (PIE) Delay Register 15 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.72 Seq0BDLY16\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY16\_pX: PHY Initialization Engine (PIE) Delay Register 16
- **Size:** 16 bits
- **Offset:** 0x90000+0x8f0+(X\*0x100000)
- **Exists:** Always



**Table 13-524 Fields for Register: Seq0BDLY16\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY16_p0	R/W	<p>Seq0BDLY16_p0: PHY Initialization Engine (PIE) Delay Register 16 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.73 Seq0BDLY17\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY17\_pX: PHY Initialization Engine (PIE) Delay Register 17
- **Size:** 16 bits
- **Offset:** 0x90000+0x8f1+(X\*0x100000)
- **Exists:** Always

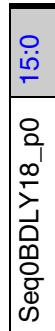


**Table 13-525 Fields for Register: Seq0BDLY17\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY17_p0	R/W	<p>Seq0BDLY17_p0: PHY Initialization Engine (PIE) Delay Register 17 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.74 Seq0BDLY18\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY18\_pX: PHY Initialization Engine (PIE) Delay Register 18
- **Size:** 16 bits
- **Offset:** 0x90000+0x8f2+(X\*0x100000)
- **Exists:** Always

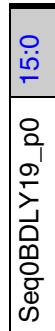


**Table 13-526 Fields for Register: Seq0BDLY18\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY18_p0	R/W	<p>Seq0BDLY18_p0: PHY Initialization Engine (PIE) Delay Register 18 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.75 Seq0BDLY19\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY19\_pX: PHY Initialization Engine (PIE) Delay Register 19
- **Size:** 16 bits
- **Offset:** 0x90000+0x8f3+(X\*0x100000)
- **Exists:** Always



**Table 13-527 Fields for Register: Seq0BDLY19\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY19_p0	R/W	<p>Seq0BDLY19_p0: PHY Initialization Engine (PIE) Delay Register 19 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.76 Seq0BDLY20\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY20\_pX: PHY Initialization Engine (PIE) Delay Register 20
- **Size:** 16 bits
- **Offset:** 0x90000+0x8f4+(X\*0x100000)
- **Exists:** Always

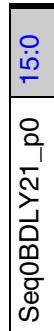


**Table 13-528 Fields for Register: Seq0BDLY20\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY20_p0	R/W	<p>Seq0BDLY20_p0: PHY Initialization Engine (PIE) Delay Register 20 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.77 Seq0BDLY21\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY21\_pX: PHY Initialization Engine (PIE) Delay Register 21
- **Size:** 16 bits
- **Offset:** 0x90000+0x8f5+(X\*0x100000)
- **Exists:** Always



**Table 13-529 Fields for Register: Seq0BDLY21\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY21_p0	R/W	<p>Seq0BDLY21_p0: PHY Initialization Engine (PIE) Delay Register 21 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.78 Seq0BDLY22\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY22\_pX: PHY Initialization Engine (PIE) Delay Register 22
- **Size:** 16 bits
- **Offset:** 0x90000+0x8f6+(X\*0x100000)
- **Exists:** Always

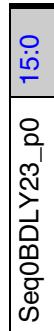


**Table 13-530 Fields for Register: Seq0BDLY22\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY22_p0	R/W	<p>Seq0BDLY22_p0: PHY Initialization Engine (PIE) Delay Register 22 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.79 Seq0BDLY23\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY23\_pX: PHY Initialization Engine (PIE) Delay Register 23
- **Size:** 16 bits
- **Offset:** 0x90000+0x8f7+(X\*0x100000)
- **Exists:** Always

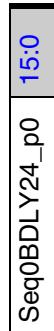


**Table 13-531 Fields for Register: Seq0BDLY23\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY23_p0	R/W	<p>Seq0BDLY23_p0: PHY Initialization Engine (PIE) Delay Register 23 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.80 Seq0BDLY24\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY24\_pX: PHY Initialization Engine (PIE) Delay Register 24
- **Size:** 16 bits
- **Offset:** 0x90000+0x8f8+(X\*0x100000)
- **Exists:** Always



**Table 13-532 Fields for Register: Seq0BDLY24\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY24_p0	R/W	<p>Seq0BDLY24_p0: PHY Initialization Engine (PIE) Delay Register 24 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.81 Seq0BDLY25\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY25\_pX: PHY Initialization Engine (PIE) Delay Register 25
- **Size:** 16 bits
- **Offset:** 0x90000+0x8f9+(X\*0x100000)
- **Exists:** Always



**Table 13-533 Fields for Register: Seq0BDLY25\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY25_p0	R/W	<p>Seq0BDLY25_p0: PHY Initialization Engine (PIE) Delay Register 25 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.82 Seq0BDLY26\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY26\_pX: PHY Initialization Engine (PIE) Delay Register 26
- **Size:** 16 bits
- **Offset:** 0x90000+0x8fa+(X\*0x100000)
- **Exists:** Always



**Table 13-534 Fields for Register: Seq0BDLY26\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY26_p0	R/W	<p>Seq0BDLY26_p0: PHY Initialization Engine (PIE) Delay Register 26 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.83 Seq0BDLY27\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY27\_pX: PHY Initialization Engine (PIE) Delay Register 27
- **Size:** 16 bits
- **Offset:** 0x90000+0x8fb+(X\*0x100000)
- **Exists:** Always



**Table 13-535 Fields for Register: Seq0BDLY27\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY27_p0	R/W	<p>Seq0BDLY27_p0: PHY Initialization Engine (PIE) Delay Register 27 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.84 Seq0BDLY28\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY28\_pX: PHY Initialization Engine (PIE) Delay Register 28
- **Size:** 16 bits
- **Offset:** 0x90000+0x8fc+(X\*0x100000)
- **Exists:** Always

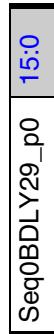


**Table 13-536 Fields for Register: Seq0BDLY28\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY28_p0	R/W	<p>Seq0BDLY28_p0: PHY Initialization Engine (PIE) Delay Register 28 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.85 Seq0BDLY29\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY29\_pX: PHY Initialization Engine (PIE) Delay Register 29
- **Size:** 16 bits
- **Offset:** 0x90000+0x8fd+(X\*0x100000)
- **Exists:** Always



**Table 13-537 Fields for Register: Seq0BDLY29\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY29_p0	R/W	<p>Seq0BDLY29_p0: PHY Initialization Engine (PIE) Delay Register 29 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.86 Seq0BDLY30\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY30\_pX: PHY Initialization Engine (PIE) Delay Register 30
- **Size:** 16 bits
- **Offset:** 0x90000+0x8fe+(X\*0x100000)
- **Exists:** Always



**Table 13-538 Fields for Register: Seq0BDLY30\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY30_p0	R/W	<p>Seq0BDLY30_p0: PHY Initialization Engine (PIE) Delay Register 30 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.87 Seq0BDLY31\_pX (for X = 0; X <= 3)

- **Description:** Seq0BDLY31\_pX: PHY Initialization Engine (PIE) Delay Register 31
- **Size:** 16 bits
- **Offset:** 0x90000+0x8ff+(X\*0x100000)
- **Exists:** Always



**Table 13-539 Fields for Register: Seq0BDLY31\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BDLY31_p0	R/W	<p>Seq0BDLY31_p0: PHY Initialization Engine (PIE) Delay Register 31 Refer to Seq0BDLY0 for full description</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.88 UpdDisFlagMask

- **Description:** UpdDisFlagMask: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 1 bit
- **Offset:** 0x90000+0x900
- **Exists:** Always



**Table 13-540 Fields for Register: UpdDisFlagMask**

Bits	Name	Memory Access	Description
0	UpdDisFlagMask	R/W	<p>UpdDisFlagMask: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.89 DisableFlagMask

- **Description:** DisableFlagMask: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x901
- **Exists:** Always

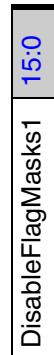


**Table 13-541 Fields for Register: DisableFlagMask**

Bits	Name	Memory Access	Description
15:0	DisableFlagMask	R/W	<p>DisableFlagMask: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.90 DisableFlagMasksX (for X = 0; X <= 1)

- **Description:** DisableFlagMasksX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x902
- **Exists:** Always



**Table 13-542 Fields for Register: DisableFlagMasksX (for X = 0; X <= 1)**

Bits	Name	Memory Access	Description
15:0	DisableFlagMasks1	R/W	<p>DisableFlagMasks1: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.91 RtrnMode\_pX (for X = 0; X <= 3)

- **Description:** RtrnMode\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 3 bits
- **Offset:** 0x90000+0x903+(X\*0x100000)
- **Exists:** Always



**Table 13-543 Fields for Register: RtrnMode\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
2:0	RtrnMode_p0	R/W	<p>RtrnMode_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.92 RtrnModeMaskX (for X = 0; X <= 1)

- **Description:** RtrnModeMaskX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x904+(X\*0x82)
- **Exists:** Always



**Table 13-544 Fields for Register: RtrnModeMaskX (for X = 0; X <= 1)**

Bits	Name	Memory Access	Description
15:0	RtrnModeMask0	R/W	<p>RtrnModeMask0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.93 RtrnModeMask0sX (for X = 0; X <= 1)

- **Description:** RtrnModeMask0sX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x905
- **Exists:** Always



**Table 13-545 Fields for Register: RtrnModeMask0sX (for X = 0; X <= 1)**

Bits	Name	Memory Access	Description
15:0	RtrnModeMask0s1	R/W	<p>RtrnModeMask0s1: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.94 CtrlUpdAck

- **Description:** CtrlUpdAck: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 1 bit
- **Offset:** 0x90000+0x912
- **Exists:** Always



**Table 13-546 Fields for Register: CtrlUpdAck**

Bits	Name	Memory Access	Description
0	CtrlUpdAck	R/W	<p>CtrlUpdAck: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.95 Seq0BResetFixedAddrBits\_pX (for X = 0; X <= 3)

- **Description:** Seq0BResetFixedAddrBits\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 1 bit
- **Offset:** 0x90000+0x938+(X\*0x100000)
- **Exists:** Always



**Table 13-547 Fields for Register: Seq0BResetFixedAddrBits\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
0	Seq0BResetFixedAddrBits_p0	R/W	<p>Seq0BResetFixedAddrBits_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.96 Seq0BCntrX (for X = 0; X <= 3)

- **Description:** Seq0BCntrX: Used by PIE during PPT2.
- **Size:** 5 bits
- **Offset:** 0x90000+0x950+(X\*0x1)
- **Exists:** Always



**Table 13-548 Fields for Register: Seq0BCntrX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
4:0	Seq0BCntr0	R/W	<p>Seq0BCntr0: Used by PIE during PPT2.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.97 Seq0BCntr0Threshold\_pX (for X = 0; X <= 3)

- **Description:** Seq0BCntr0Threshold\_pX: Used by PIE during PPT2.
- **Size:** 5 bits
- **Offset:** 0x90000+0x954+(X\*0x100000)
- **Exists:** Always



**Table 13-549 Fields for Register: Seq0BCntr0Threshold\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
4:0	Seq0BCntr0Threshold_p0	R/W	<p>Seq0BCntr0Threshold_p0: Used by PIE during PPT2.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.98 Seq0BCntr1Threshold\_pX (for X = 0; X <= 3)

- **Description:** Seq0BCntr1Threshold\_pX: Used by PIE during PPT2.
- **Size:** 5 bits
- **Offset:** 0x90000+0x955+(X\*0x100000)
- **Exists:** Always



**Table 13-550 Fields for Register: Seq0BCntr1Threshold\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
4:0	Seq0BCntr1Threshold_p0	R/W	<p>Seq0BCntr1Threshold_p0: Used by PIE during PPT2.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.99 Seq0BCntr2Threshold\_pX (for X = 0; X <= 3)

- **Description:** Seq0BCntr2Threshold\_pX: Used by PIE during PPT2.
- **Size:** 5 bits
- **Offset:** 0x90000+0x956+(X\*0x100000)
- **Exists:** Always



**Table 13-551 Fields for Register: Seq0BCntr2Threshold\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
4:0	Seq0BCntr2Threshold_p0	R/W	<p>Seq0BCntr2Threshold_p0: Used by PIE during PPT2.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.100 Seq0BCntr3Threshold\_pX (for X = 0; X <= 3)

- **Description:** Seq0BCntr3Threshold\_pX: Used by PIE during PPT2.
- **Size:** 5 bits
- **Offset:** 0x90000+0x957+(X\*0x100000)
- **Exists:** Always



**Table 13-552 Fields for Register: Seq0BCntr3Threshold\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
4:0	Seq0BCntr3Threshold_p0	R/W	<p>Seq0BCntr3Threshold_p0: Used by PIE during PPT2.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.101 Seq0BCntrCtrl

- **Description:** Seq0BCntrCtrl: Used by PIE during PPT2.
- **Size:** 8 bits
- **Offset:** 0x90000+0x958
- **Exists:** Always



Table 13-553 Fields for Register: Seq0BCntrCtrl

Bits	Name	Memory Access	Description
7:0	Seq0BCntrCtrl	R/W	Seq0BCntrCtrl: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.5.102 RtrnModeMask1sX (for X = 0; X <= 1)

- **Description:** RtrnModeMask1sX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0x987
- **Exists:** Always

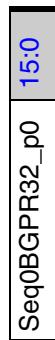


**Table 13-554 Fields for Register: RtrnModeMask1sX (for X = 0; X <= 1)**

Bits	Name	Memory Access	Description
15:0	RtrnModeMask1s1	R/W	<p>RtrnModeMask1s1: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.103 Seq0BGPR32\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR32\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0xa20+(X\*0x100000)
- **Exists:** Always

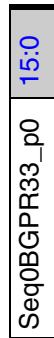


**Table 13-555 Fields for Register: Seq0BGPR32\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR32_p0	R/W	<p>Seq0BGPR32_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.104 Seq0BGPR33\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR33\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0xa21+(X\*0x100000)
- **Exists:** Always

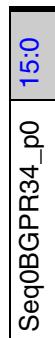


**Table 13-556 Fields for Register: Seq0BGPR33\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR33_p0	R/W	<p>Seq0BGPR33_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.105 Seq0BGPR34\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR34\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0xa22+(X\*0x100000)
- **Exists:** Always



**Table 13-557 Fields for Register: Seq0BGPR34\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR34_p0	R/W	<p>Seq0BGPR34_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.106 Seq0BGPR35\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR35\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0xa23+(X\*0x100000)
- **Exists:** Always

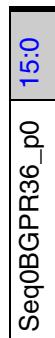


**Table 13-558 Fields for Register: Seq0BGPR35\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR35_p0	R/W	<p>Seq0BGPR35_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.107 Seq0BGPR36\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR36\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0xa24+(X\*0x100000)
- **Exists:** Always

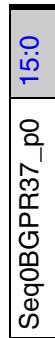


**Table 13-559 Fields for Register: Seq0BGPR36\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR36_p0	R/W	<p>Seq0BGPR36_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.108 Seq0BGPR37\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR37\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0xa25+(X\*0x100000)
- **Exists:** Always

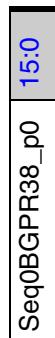


**Table 13-560 Fields for Register: Seq0BGPR37\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR37_p0	R/W	<p>Seq0BGPR37_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.109 Seq0BGPR38\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR38\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0xa26+(X\*0x100000)
- **Exists:** Always



**Table 13-561 Fields for Register: Seq0BGPR38\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR38_p0	R/W	<p>Seq0BGPR38_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.110 Seq0BGPR39\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR39\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0xa27+(X\*0x100000)
- **Exists:** Always



**Table 13-562 Fields for Register: Seq0BGPR39\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR39_p0	R/W	<p>Seq0BGPR39_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.111 Seq0BGPR40\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR40\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0xa28+(X\*0x100000)
- **Exists:** Always

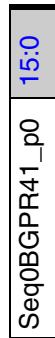


**Table 13-563 Fields for Register: Seq0BGPR40\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR40_p0	R/W	<p>Seq0BGPR40_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.112 Seq0BGPR41\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR41\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0xa29+(X\*0x100000)
- **Exists:** Always

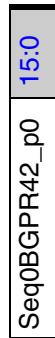


**Table 13-564 Fields for Register: Seq0BGPR41\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR41_p0	R/W	<p>Seq0BGPR41_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.113 Seq0BGPR42\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR42\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0xa2a+(X\*0x100000)
- **Exists:** Always

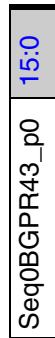


**Table 13-565 Fields for Register: Seq0BGPR42\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR42_p0	R/W	<p>Seq0BGPR42_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.114 Seq0BGPR43\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR43\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0xa2b+(X\*0x100000)
- **Exists:** Always



**Table 13-566 Fields for Register: Seq0BGPR43\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR43_p0	R/W	<p>Seq0BGPR43_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.115 Seq0BGPR44\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR44\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0xa2c+(X\*0x100000)
- **Exists:** Always



**Table 13-567 Fields for Register: Seq0BGPR44\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR44_p0	R/W	<p>Seq0BGPR44_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.116 Seq0BGPR45\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR45\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0xa2d+(X\*0x100000)
- **Exists:** Always

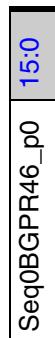


**Table 13-568 Fields for Register: Seq0BGPR45\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR45_p0	R/W	<p>Seq0BGPR45_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.117 Seq0BGPR46\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR46\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0xa2e+(X\*0x100000)
- **Exists:** Always

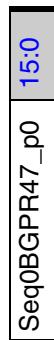


**Table 13-569 Fields for Register: Seq0BGPR46\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR46_p0	R/W	<p>Seq0BGPR46_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.5.118 Seq0BGPR47\_pX (for X = 0; X <= 3)

- **Description:** Seq0BGPR47\_pX: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0x90000+0xa2f+(X\*0x100000)
- **Exists:** Always



**Table 13-570 Fields for Register: Seq0BGPR47\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
15:0	Seq0BGPR47_p0	R/W	<p>Seq0BGPR47_p0: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

## 13.6 **DWC\_DDRPHYA\_ZCALj\_Pk Registers**

### 13.6.1 ZcalClkDiv

- **Description:** ZcalClkDiv: Power-saving clock divider to ZCAL
- **Size:** 2 bits
- **Offset:** 0xb0000+0x1
- **Exists:** Always



**Table 13-571 Fields for Register: ZcalClkDiv**

Bits	Name	Memory Access	Description
1:0	ZcalClkDiv	R/W	<p>ZcalClkDiv: Prior to CSR access in HMZCAL, csrZcalClkDiv should be set to 0 00 Recommended for DfiClk &lt; 800 MHz 01 Debug purpose only 10 Debug purpose only 11 Recommended for DfiClk &gt;=800 MHz In HMZCAL, When ZcalStopClk=1, this csr can not be accessed.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.6.2 ZCalClkInfo\_pX (for X = 0; X <= 3)

- **Name:** Impedance Calibration Clock Ratio
- **Description:** ZCalClkInfo\_pX: Informs the impedance calibration engine about frequency of DfiClk
- **Size:** 11 bits
- **Offset:** 0xb0000+0x4+(X\*0x100000)
- **Exists:** Always



**Table 13-572 Fields for Register: ZCalClkInfo\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
10:0	ZCalDfiClkTicksPer1uS	R/W	<p>ZCalDfiClkTicksPer1uS: Must be programmed to the number of DfiClks in 1us (rounded up), with minimum value of 24.</p> <ul style="list-style-type: none"> <li>■ if (DfiClk &lt; 24MHz) ZCalDfiClkTicksPer1uS = 24</li> <li>■ else ZCalDfiClkTicksPer1uS = (number of DfiClks in 1us)</li> </ul> <p><b>Value After Reset:</b> 0x320</p> <p><b>Exists:</b> Always</p>

### 13.6.3 MtestMuxSel

- **Description:** MtestMuxSel: Digital Observation Pin control
- **Size:** 10 bits
- **Offset:** 0xb0000+0x1a
- **Exists:** Always



**Table 13-573 Fields for Register: MtestMuxSel**

Bits	Name	Memory Access	Description
9:0	MtestMuxSel	R/W	<p>MtestMuxSel: Controls for the mux for asynchronous data to the Digital Observation Pin.</p> <ul style="list-style-type: none"> <li>■ Encoding 9'h0 causes this chiplet to drive 0, (allowing flat 'OR' of pass-through information).</li> <li>■ MtestMuxSel[4:0] - Lower 5 bits selects one bit from the 32 bit MtestMux in each section or slice.</li> </ul> <p>Detailed tables are in the PUB Databook for PUB sections (AC, MASTER, etc.)</p> <ul style="list-style-type: none"> <li>■ MtestMuxSel[8:5] - Where more than one MtestMux exists, non-zero values select the outputs of the additional Mtest-Muxes,           <ul style="list-style-type: none"> <li>□ DBYTE MtestMuxSel[6:5]=2'h0 for Mux-A and 2'h1 for Mux-B and 2'h2 for Mux-C.</li> <li>□ For all other slaves, MtestMuxSel[8:5] are unused</li> </ul> </li> </ul> <p><b>Note:</b> See the PUB Databook for how, or if, the Digital Observation Pin is mapped to a physical bump in this configuration.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.6.4 ZCALParityInvert

- **Description:** ZCALParityInvert: Invert APB Parity for register slave ZCAL
- **Size:** 2 bits
- **Offset:** 0xb0000+0x4d
- **Exists:** Always



**Table 13-574 Fields for Register: ZCALParityInvert**

Bits	Name	Memory Access	Description
1:0	ZCALParityInvert	R/W	<p>ZCALParityInvert: Invert APB Parity for register slave ZCAL. As required for Automotive. NOTE: This register should be used only for test. Set the bits for only one slave at a time. When bits are set for a particular slave. APB Reads of only that slave are valid. Bit 0 applies to [7:0] Bit 1 applies to [15:8] In HMZCAL, When ZcalStopClk=1, this csr can not be accessed.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.6.5 ScratchPadZCAL

- **Description:** ScratchPadZCAL: ScratchPad for ZCAL
- **Size:** 16 bits
- **Offset:** 0xb0000+0x7d
- **Exists:** Always



**Table 13-575 Fields for Register: ScratchPadZCAL**

Bits	Name	Memory Access	Description
15:0	ScratchPadZCAL	R/W	<p>ScratchPadZCAL: ScratchPad for ZCAL. As required for Automotive. In HMZCAL, When ZcalStopClk=1, this csr can not be accessed.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.6.6 ZCALReservedX (for X == 0)

- **Description:** ZCALReservedX: Reserved for future use
- **Size:** 16 bits
- **Offset:** 0xb0000+0xfc
- **Exists:** Always



**Table 13-576 Fields for Register: ZCALReservedX (for X == 0)**

Bits	Name	Memory Access	Description
15:0	ZCALReserved0	R/W	ZCALReserved0: Reserved for future use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.6.7 PUBReservedP1\_pX (ZCAL) (for X = 0; X <= 3)

- **Description:** PUBReservedP1\_pX: Reserved for future use
- **Size:** 8 bits
- **Offset:** 0xb0000+0xff+(X\*0x100000)
- **Exists:** Always

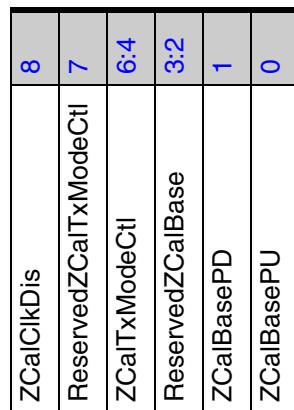


**Table 13-577 Fields for Register: PUBReservedP1\_pX (ZCAL) (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
7:0	PUBReservedP1_p0	R/W	PUBReservedP1_p0: Reserved for future use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.6.8 ZCalBaseCtrl

- **Name:** Impedance Calibration control
- **Description:** ZCalBaseCtrl: Base legs control for the SE IOs used for Impedance Calibration. TX Mode Control for the SE IOs used for Impedance Calibration. Reserved CSRs to support debug and non-mission-mode features.
- **Size:** 9 bits
- **Offset:** 0xb0000+0x301
- **Exists:** Always



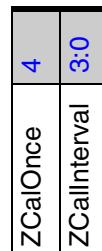
**Table 13-578 Fields for Register: ZCalBaseCtrl**

Bits	Name	Memory Access	Description
8	ZCalClkDis	R/W	ZCalClkDis: Reserved. Must be 1'b0 for correct operation of the ZCal Logic. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7	ReservedZCalTxModeCtl	R/W	ReservedZCalTxModeCtl: Reserved for future use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
6:4	ZCalTxModeCtl	R/W	ZCalTxModeCtl: Impedance calibration Tx Mode Control. Refer to Technology specific PHY databook Custom CKT Macro Optimal Settings for any recommended settings, as available <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
3:2	ReservedZCalBase	R/W	ReservedZCalBase: Reserved for future use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

Bits	Name	Memory Access	Description
1	ZCalBasePD	R/W	ZCalBasePD: Reserved. Must be 0. <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always
0	ZCalBasePU	R/W	ZCalBasePU: Reserved. Must be 0. <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always

### 13.6.9 ZCalRate

- **Name:** Impedance Calibration timing control
- **Description:** ZCalRate: This CSR controls the impedance calibration timings in the PHY.
- **Size:** 5 bits
- **Offset:** 0xb0000+0x303
- **Exists:** Always



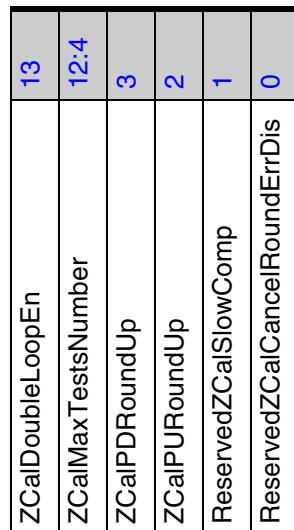
**Table 13-579 Fields for Register: ZCalRate**

Bits	Name	Memory Access	Description
4	ZCalOnce	R/W	<p>ZCalOnce: The setting of this CSR changes the behavior of CSR ZCalRun.</p> <ul style="list-style-type: none"> <li>■ 1: A 0-&gt;1 transition of CSR ZCalRun causes a single iteration of the calibration sequence to occur. When ZCalRun=1, a 1-&gt;0 transition in ZCalReset will also cause a single iteration of the calibration sequence to occur. Once the calibration is complete, the calibration engine will remain IDLE until another Trigger.</li> <li>■ 0: Calibration will proceed at the rate determined by Register ZCallInterval.</li> </ul> <p>This field should only be changed while the calibrator is idle i.e., when ZCalReset=1.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
3:0	ZCallInterval	R/W	<p>ZCallInterval: This CSR specifies the interval between successive calibrations, in mS.</p> <ul style="list-style-type: none"> <li>■ 0: continuous (minimum 1us)</li> <li>■ 1: Reserved</li> <li>■ 2: Reserved</li> <li>■ 3: 1 ms</li> <li>■ 4: 2 ms</li> <li>■ 5: 3 ms</li> <li>■ 6: 4 ms</li> <li>■ 7: 8 ms</li> <li>■ 8: 10 ms</li> <li>■ 9: 20 ms</li> <li>■ 10-15: Reserved</li> </ul> <p>This field should only be changed while the calibrator is idle. i.e., when csr ZCalReset=1. The calibration intervals are valid only at DFICLK frequency &gt;=24MHz. If the DFICLK frequency is less than 24MHz, the interval may be up to 2.4x larger than the value programmed in the register.</p> <p><b>Value After Reset:</b> 0x9</p> <p><b>Exists:</b> Always</p>

### 13.6.10 ZCalCtrl

- **Name:** Code Search related impedance calibration controls
- **Description:** ZCalCtrl: Code Search related impedance calibration controls.
- **Size:** 14 bits
- **Offset:** 0xb0000+0x305
- **Exists:** Always



**Table 13-580 Fields for Register: ZCalCtrl**

Bits	Name	Memory Access	Description
13	ZCalDoubleLoopEn	R/W	<p>ZCalDoubleLoopEn: When set, enable double-loop calibration in the first calibration run.</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>
12:4	ZCalMaxTestsNumber	R/W	<p>ZCalMaxTestsNumber: Maximum number of tests for the code search algorithm to converge.</p> <p><b>Value After Reset:</b> 0x1ff</p> <p><b>Exists:</b> Always</p>
3	ZCalPDRoundUp	R/W	<p>ZCalPDRoundUp: When set, enable RoundUp feature for PD calibration.</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>
2	ZCalPURoundUp	R/W	<p>ZCalPURoundUp: When set, enable RoundUp feature for PU calibration.</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
1	ReservedZCalSlowComp	R/W	ReservedZCalSlowComp: Reserved. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
0	ReservedZCalCancelRoundErrDis	R/W	ReservedZCalCancelRoundErrDis: Reserved <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always

### 13.6.11 ZCalComplInvert

- **Name:** Impedance Calibration Comparator Invert control
- **Description:** ZCalComplInvert: Safety bits in the calibration targets, which enables the inversion of the comparator output before it is sampled. A 1 in the CSR field causes the comparator output to be inverted.
- **Size:** 3 bits
- **Offset:** 0xb0000+0x306
- **Exists:** Always



**Table 13-581 Fields for Register: ZCalComplInvert**

Bits	Name	Memory Access	Description
2	ZCalComplInvertPD	R/W	ZCalComplInvertPD: When set, inverts the comparator output when Pull-Down Calibration is running <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always
1	ZCalComplInvertPU	R/W	ZCalComplInvertPU: When set, inverts the comparator output when Pull-up Calibration is running <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
0	ZCalComplInvertComp	R/W	ZCalComplInvertComp: When set, inverts the comparator output when Comparator Offset Calibration is running <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always

### 13.6.12 ZCalCompOffset

- **Name:** Impedance Calibration Comparator offset control
- **Description:** ZCalCompOffset: A safety register that can be used to correct systematic offsets from comparator calibration stage. It is not recommended to change this value from the default zero value.
- **Size:** 6 bits
- **Offset:** 0xb0000+0x307
- **Exists:** Always



**Table 13-582 Fields for Register: ZCalCompOffset**

Bits	Name	Memory Access	Description
5:0	ZCalCompOffsetVal	R/W	<p>ZCalCompOffsetVal: This value adjusts the comparator offset calibration result 2's complement coded, to specify positive or negative offset values.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.6.13 ZCalPUOffset

- **Name:** Impedance Calibration Pull-up offset control
- **Description:** ZCalPUOffset: A safety register that can be used to correct systematic offsets from pull-up calibration stage. It is not recommended to change this value from the default zero value.
- **Size:** 4 bits
- **Offset:** 0xb0000+0x308
- **Exists:** Always



**Table 13-583 Fields for Register: ZCalPUOffset**

Bits	Name	Memory Access	Description
3:0	ZCalPUOffsetVal	R/W	<p>ZCalPUOffsetVal: This value adjusts the driver pull-up calibration code 2's complement coded, to specify positive or negative offset values.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.6.14 ZCalCompOvr

- **Name:** Impedance Calibrator's comparator calibration override control
- **Description:** ZCalCompOvr: Control the override of the comparator offset calibration results. This CSR may only be written when the calibrator is not running.
- **Size:** 16 bits
- **Offset:** 0xb0000+0x309
- **Exists:** Always

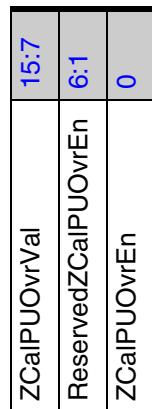


**Table 13-584 Fields for Register: ZCalCompOvr**

Bits	Name	Memory Access	Description
15	ReservedZCalCompOvrVal	R/W	ReservedZCalCompOvrVal: Reserved for future use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
14:8	ZCalCompOvrVal	R/W	ZCalCompOvrVal: If the CSR ZCalCompOvrEn is set, then the value provided here by software will be used for pull-up and pull-down calibration, instead of the value obtained from the Comparator calibration. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7:1	ReservedZCalCompOvrEn	R/W	ReservedZCalCompOvrEn: Reserved for future use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
0	ZCalCompOvrEn	R/W	ZCalCompOvrEn: 1'b1 - Enable the override of the comparator offset calibration result <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.6.15 ZCalPUOvr

- **Name:** Impedance Calibrator's pull-up calibration override control
- **Description:** ZCalPUOvr: Control the override of the pull-up calibration results. This CSR may only be written when the calibrator is not running.
- **Size:** 16 bits
- **Offset:** 0xb0000+0x30a
- **Exists:** Always



**Table 13-585 Fields for Register: ZCalPUOvr**

Bits	Name	Memory Access	Description
15:7	ZCalPUOvrVal	R/W	<p>ZCalPUOvrVal: If the CSR ZCalPUOvrEn is set, then the value provided here by software will be used for pull-up calibration and propagated out as the reference pull-up code, instead of the value obtained from the pull-up calibration.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
6:1	ReservedZCalPUOvrEn	R/W	<p>ReservedZCalPUOvrEn: Reserved for future use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
0	ZCalPUOvrEn	R/W	<p>ZCalPUOvrEn: 1'b1 - Enable the override of the pull-up calibration result</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.6.16 ZCalClrFirstRunDone

- **Description:** ZCalClrFirstRunDone: Clear FirstRunDone flag, inside Impedance Calibration control logic
- **Size:** 1 bit
- **Offset:** 0xb0000+0x30d
- **Exists:** Always



**Table 13-586 Fields for Register: ZCalClrFirstRunDone**

Bits	Name	Memory Access	Description
0	ZCalClrFirstRunDone	R/W	<p>ZCalClrFirstRunDone: Asserting this CSR clear the FirstRunDone flag. FirstRunDone is 0 after reset and will go 1 every time one calibration run is complete.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.6.17 ZCalPDSearchGainTV

- **Description:** ZCalPDSearchGainTV: Search code end-gain control for pull-down calibration
- **Size:** 9 bits
- **Offset:** 0xb0000+0x30e
- **Exists:** Always



**Table 13-587 Fields for Register: ZCalPDSearchGainTV**

Bits	Name	Memory Access	Description
8	ZCalPDSearchGainTVAuto	R/W	<p>ZCalPDSearchGainTVAuto: Automatic gain change after first calibration. If asserted, the first pull-down calibration will end the code search using the gain specified in CSR ZCalPDSearchGainTVVal for the first calibration; and use the gain specified in CSR ZCalPDSearchGainTVValB for the second and subsequent calibrations.</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>
7:4	ZCalPDSearchGainTVValB	R/W	<p>ZCalPDSearchGainTVValB: End-gain value to end the code search. This value is never used for the first calibration. It is used for second and subsequent calibrations when ZCalPDSearchGainTVAuto is 1.</p> <p>The gain value encoding is the same used for ZCalPDSearchGainTVVal.</p> <p><b>Value After Reset:</b> 0x8</p> <p><b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
3:0	ZCalPDSearchGainTVVal	R/W	<p>ZCalPDSearchGainTVVal: End-gain value to end the code search. When the search algorithm get down to this gain value, as soon as the test result toggles, the search ends. This value is used for the first calibration and will be used for the subsequent ones only if ZCalPDSearchGainTVAuto is 0. The gain value encoding will result in the step deltas below:</p> <ul style="list-style-type: none"> <li>■ 4'h0 - Delta = 32.0</li> <li>■ 4'h1 - Delta = 16.0</li> <li>■ 4'h2 - Delta = 8.0</li> <li>■ 4'h3 - Delta = 4.0</li> <li>■ 4'h4 - Delta = 2.0</li> <li>■ 4'h5 - Delta = 1.0</li> <li>■ 4'h6 - Delta = 0.5</li> <li>■ 4'h7 - Delta = 0.25</li> <li>■ 4'h8 - Delta = 0.125</li> <li>■ 4'h9 - Delta = 0.0625</li> <li>■ 4'hA - Delta = 0.03125</li> <li>■ 4'hB - Delta = 0.015625</li> <li>■ 4'hC to 4'hF - Invalid</li> </ul> <p><b>Value After Reset:</b> 0x8</p> <p><b>Exists:</b> Always</p>

### 13.6.18 ZCalAnaSettlingTime

- **Description:** ZCalAnaSettlingTime: ZCalana settling time
- **Size:** 6 bits
- **Offset:** 0xb0000+0x30f
- **Exists:** Always



**Table 13-588 Fields for Register: ZCalAnaSettlingTime**

Bits	Name	Memory Access	Description
5:0	ZCalAnaSettlingTime	R/W	<p>ZCalAnaSettlingTime: Variable to specify the time ZCalana's comparator output takes to settle, after a code change in SEPU, SEPD or VOH DAC that will affect the voltage level at the comparator inputs. For all VDD values , ZCalAnaSettlingTime = 1Dh</p> <p><b>Value After Reset:</b> 0x10</p> <p><b>Exists:</b> Always</p>

### 13.6.19 ZCalReset

- **Description:** ZCalReset: Impedance Calibration Off/Reset
- **Size:** 1 bit
- **Offset:** 0xb0000+0x310
- **Exists:** Always



**Table 13-589 Fields for Register: ZCalReset**

Bits	Name	Memory Access	Description
0	ZCalReset	R/W	<p>ZCalReset: This CSR is written by PHY Initialization Engine (PIE) and the data in here will be overwritten. Asserting this csr resets the calibrator to its idle state. Must be cleared in order for the calibration engine to run again. When this CSR is deasserted, one calibration run starts automatically, if ZCalRun=1.</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

### 13.6.20 ZCalRun

- **Description:** ZCalRun: Impedance Calibration trigger
- **Size:** 1 bit
- **Offset:** 0xb0000+0x311
- **Exists:** Always



**Table 13-590 Fields for Register: ZCalRun**

Bits	Name	Memory Access	Description
0	ZCalRun	R/W	<p>ZCalRun: This CSR triggers the impedance calibration sequence.</p> <ul style="list-style-type: none"> <li>■ A calibration sequence will be triggered by the 0-&gt;1 transition of this bit, as determined by CSR ZCalOnce.</li> </ul> <p>If ZCalRun=0, the calibrator will not run. ZcalRun should not be set to 0 during active calibration. This csr will not do anything while ZCalReset is asserted.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.6.21 ZCalBusy

- **Description:** ZCalBusy: Impedance Calibration Busy Status
- **Size:** 1 bit
- **Offset:** 0xb0000+0x312
- **Exists:** Always



**Table 13-591 Fields for Register: ZCalBusy**

Bits	Name	Memory Access	Description
0	ZCalBusy	R	<p>ZCalBusy: Read 1 if the calibrator is actively calibrating. Any changes to calibrator-related CSRs may only be made if the calibrator is disabled (via CSR ZCalRun) and this CSR reads 0.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.6.22 ZCalCompResult

- **Description:** ZCalCompResult: Impedance Calibrator comparator calibration code monitor
- **Size:** 7 bits
- **Offset:** 0xb0000+0x313
- **Exists:** Always



**Table 13-592 Fields for Register: ZCalCompResult**

Bits	Name	Memory Access	Description
6:0	ZCalCompResult	R	<p>ZCalCompResult: For ZCal debug only</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.6.23 ZCalPUResult

- **Description:** ZCalPUResult: Impedance Calibrator pull-up calibration code monitor
- **Size:** 9 bits
- **Offset:** 0xb0000+0x314
- **Exists:** Always



**Table 13-593 Fields for Register: ZCalPUResult**

Bits	Name	Memory Access	Description
8:0	ZCalPUResult	R	<p>ZCalPUResult: For ZCal debug only</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.6.24 ZCalPDResult

- **Description:** ZCalPDResult: Impedance Calibrator pull-down calibration code monitor
- **Size:** 9 bits
- **Offset:** 0xb0000+0x315
- **Exists:** Always



Table 13-594 Fields for Register: ZCalPDResult

Bits	Name	Memory Access	Description
8:0	ZCalPDResult	R	<p>ZCalPDResult: For ZCal debug only</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.6.25 ZCalCodePU

- **Description:** ZCalCodePU: Pull-up impedance calibration code from the calibration circuit
- **Size:** 9 bits
- **Offset:** 0xb0000+0x316
- **Exists:** Always



**Table 13-595 Fields for Register: ZCalCodePU**

Bits	Name	Memory Access	Description
8:0	ZCalCodePU	R	<p>ZCalCodePU: Latest Pull-up impedance calibration code saved. Ready to go to the IOs after an update event. The value reaching the IO will also depend on ZQCalOffsetPU and ZQCalCodeOvrEnPU values. Refer to ZQCalCodePU description.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.6.26 ZCalCodePD

- **Description:** ZCalCodePD: Pull-down impedance calibration code from the calibration circuit
- **Size:** 9 bits
- **Offset:** 0xb0000+0x317
- **Exists:** Always

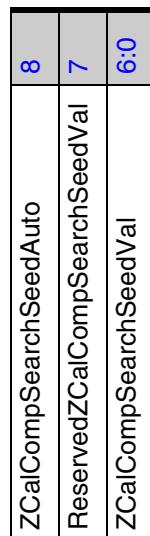


**Table 13-596 Fields for Register: ZCalCodePD**

Bits	Name	Memory Access	Description
8:0	ZCalCodePD	R	<p>ZCalCodePD: Latest Pull-down impedance calibration code saved. Ready to go to the IOs after an update event. The value reaching the IO will also depend on ZQCalOffsetPD and ZQCalCodeOvrEnPD values. Refer to ZQCalCodePD description.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.6.27 ZCalCompSearchSeed

- **Description:** ZCalCompSearchSeed: Seed control for comparator offset calibration
- **Size:** 9 bits
- **Offset:** 0xb0000+0x318
- **Exists:** Always



**Table 13-597 Fields for Register: ZCalCompSearchSeed**

Bits	Name	Memory Access	Description
8	ZCalCompSearchSeedAuto	R/W	<p>ZCalCompSearchSeedAuto: Automatic seed update between calibrations. If asserted, the first comparator offset calibration will use the seed value specified in CSR ZCalCompSearchSeedVal; and the next calibrations will use as seed value, the value from the previous calibration.</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>
7	ReservedZCalCompSearchSeedVal	R/W	<p>ReservedZCalCompSearchSeedVal: Reserved for future use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
6:0	ZCalCompSearchSeedVal	R/W	<p>ZCalCompSearchSeedVal: comparator offset calibration seed value</p> <p><b>Value After Reset:</b> 0x40</p> <p><b>Exists:</b> Always</p>

### 13.6.28 ZCalPUSearchSeed

- **Description:** ZCalPUSearchSeed: Seed control for pull-up calibration
- **Size:** 10 bits
- **Offset:** 0xb0000+0x319
- **Exists:** Always



**Table 13-598 Fields for Register: ZCalPUSearchSeed**

Bits	Name	Memory Access	Description
9	ZCalPUSearchSeedAuto	R/W	<p>ZCalPUSearchSeedAuto: Automatic seed update between calibrations. If asserted, the first pull-up calibration will use the seed value specified in CSR ZCalPUSearchSeedVal; and the next calibrations will use as seed value, the value from the previous calibration.</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>
8:0	ZCalPUSearchSeedVal	R/W	<p>ZCalPUSearchSeedVal: pull-up calibration seed value</p> <p><b>Value After Reset:</b> 0x80</p> <p><b>Exists:</b> Always</p>

### 13.6.29 ZCalPDSearchSeed

- **Description:** ZCalPDSearchSeed: Seed control for pull-down calibration
- **Size:** 10 bits
- **Offset:** 0xb0000+0x31a
- **Exists:** Always



**Table 13-599 Fields for Register: ZCalPDSearchSeed**

Bits	Name	Memory Access	Description
9	ZCalPDSearchSeedAuto	R/W	<p>ZCalPDSearchSeedAuto: Automatic seed update between calibrations. If asserted, the first pull-down calibration will use the seed value specified in CSR ZCalPDSearchSeedVal; and the next calibrations will use as seed value, the value from the previous calibration.</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>
8:0	ZCalPDSearchSeedVal	R/W	<p>ZCalPDSearchSeedVal: pull-down calibration seed value</p> <p><b>Value After Reset:</b> 0x80</p> <p><b>Exists:</b> Always</p>

### 13.6.30 ZCalCompSearchGainIV

- **Description:** ZCalCompSearchGainIV: Search code initial-gain control for comparator offset calibration
- **Size:** 9 bits
- **Offset:** 0xb0000+0x31b
- **Exists:** Always



**Table 13-600 Fields for Register: ZCalCompSearchGainIV**

Bits	Name	Memory Access	Description
8	ZCalCompSearchGainIVAuto	R/W	<p>ZCalCompSearchGainIVAuto: Automatic gain change after first calibration. If asserted, the first comparator offset calibration will start the code search using the gain specified in CSR ZCalCompSearchGainIVVal for the first calibration; and use the gain specified in CSR ZCalCompSearchGainIVValB for the second and subsequent calibrations</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>
7:4	ZCalCompSearchGainIVValB	R/W	<p>ZCalCompSearchGainIVValB: Initial-gain value to start code search with. This value is never used for the first calibration. It is used for the second and subsequent calibrations when ZCalCompSearchGainIVAuto is 1. The gain value encoding is the same used for ZCalCompSearchGainIVVal.</p> <p><b>Value After Reset:</b> 0x3</p> <p><b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
3:0	ZCalCompSearchGainIVVal	R/W	<p>ZCalCompSearchGainIVVal: Initial-gain value to start code search with. This value is used for the first calibration and will be used for the subsequent ones only if ZCalCompSearchGainIVAuto is 0. The gain value encoding will result in the step deltas below:</p> <ul style="list-style-type: none"> <li>■ 4'h0 - Delta = 8.0</li> <li>■ 4'h1 - Delta = 4.0</li> <li>■ 4'h2 - Delta = 2.0</li> <li>■ 4'h3 - Delta = 1.0</li> <li>■ 4'h4 - Delta = 0.5</li> <li>■ 4'h5 - Delta = 0.25</li> <li>■ 4'h6 - Delta = 0.125</li> <li>■ 4'h7 - Delta = 0.0625</li> <li>■ 4'h8 - Delta = 0.03125</li> <li>■ 4'h9 - Delta = 0.015625</li> <li>■ 4'hA - Delta = 0.007814</li> <li>■ 4'hB - Delta = 0.003907</li> <li>■ 4'hC to 4'hF - Invalid</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.6.31 ZCalCompSearchGainTV

- **Description:** ZCalCompSearchGainTV: Search code end-gain control for comparator offset calibration
- **Size:** 9 bits
- **Offset:** 0xb0000+0x31c
- **Exists:** Always



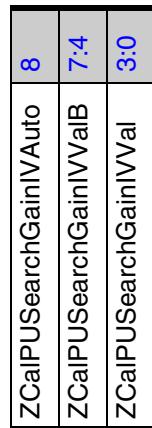
**Table 13-601 Fields for Register: ZCalCompSearchGainTV**

Bits	Name	Memory Access	Description
8	ZCalCompSearchGainTVAuto	R/W	ZCalCompSearchGainTVAuto: Automatic gain change after first calibration. If asserted, the first comparator offset calibration will end the code search using the gain specified in CSR ZCalCompSearchGainTVVal for the first calibration; and use the gain specified in CSR ZCalCompSearchGainTVValB for the second and subsequent calibrations <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always
7:4	ZCalCompSearchGainTVValB	R/W	ZCalCompSearchGainTVValB: End-gain value to end the code search. This value is never used for the first calibration. It is used for second and subsequent calibrations when ZCalCompSearchGainTVAuto is 1. The gain value encoding is the same used for ZCalCompSearchGainTVVal <b>Value After Reset:</b> 0x8 <b>Exists:</b> Always

Bits	Name	Memory Access	Description
3:0	ZCalCompSearchGainTVVal	R/W	<p>ZCalCompSearchGainTVVal: End-gain value to end the code search. When the search algorithm get down to this gain value, as soon as the test result toggles, the search ends. This value is used for the first calibration and will be used for the subsequent ones only if ZCalCompSearchGainTVAuto is 0. The gain value encoding will result in the step deltas below:</p> <ul style="list-style-type: none"> <li>■ 4'h0 - Delta = 8.0</li> <li>■ 4'h1 - Delta = 4.0</li> <li>■ 4'h2 - Delta = 2.0</li> <li>■ 4'h3 - Delta = 1.0</li> <li>■ 4'h4 - Delta = 0.5</li> <li>■ 4'h5 - Delta = 0.25</li> <li>■ 4'h6 - Delta = 0.125</li> <li>■ 4'h7 - Delta = 0.0625</li> <li>■ 4'h8 - Delta = 0.03125</li> <li>■ 4'h9 - Delta = 0.015625</li> <li>■ 4'hA - Delta = 0.007814</li> <li>■ 4'hB - Delta = 0.003907</li> <li>■ 4'hC to 4'hF - Invalid</li> </ul> <p><b>Value After Reset:</b> 0x8</p> <p><b>Exists:</b> Always</p>

### 13.6.32 ZCalPUSearchGainIV

- **Description:** ZCalPUSearchGainIV: Search code initial-gain control for pull-up calibration
- **Size:** 9 bits
- **Offset:** 0xb0000+0x31d
- **Exists:** Always



**Table 13-602 Fields for Register: ZCalPUSearchGainIV**

Bits	Name	Memory Access	Description
8	ZCalPUSearchGainIVAuto	R/W	<p>ZCalPUSearchGainIVAuto: Automatic gain change after first calibration. If asserted, the first pull-up calibration will start the code search using the gain specified in CSR ZCalPUSearchGainIVVal for the first calibration; and use the gain specified in CSR ZCalPUSearchGainIVValB for the second and subsequent calibrations</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>
7:4	ZCalPUSearchGainIVValB	R/W	<p>ZCalPUSearchGainIVValB: Initial-gain value to start code search with. This value is never used for the first calibration. It is used for the second and subsequent calibrations when ZCalPUSearchGainIVAuto is 1. The gain value encoding is the same used for ZCalPUSearchGainIVVal.</p> <p><b>Value After Reset:</b> 0x3</p> <p><b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
3:0	ZCalPUSearchGainIVVal	R/W	<p>ZCalPUSearchGainIVVal: Initial-gain value to start code search with. This value is used for the first calibration and will be used for the subsequent ones only if ZCalPUSearchGainIVAuto is 0. The gain value encoding will result in the step deltas below:</p> <ul style="list-style-type: none"> <li>■ 4'h0 - Delta = 32.0</li> <li>■ 4'h1 - Delta = 16.0</li> <li>■ 4'h2 - Delta = 8.0</li> <li>■ 4'h3 - Delta = 4.0</li> <li>■ 4'h4 - Delta = 2.0</li> <li>■ 4'h5 - Delta = 1.0</li> <li>■ 4'h6 - Delta = 0.5</li> <li>■ 4'h7 - Delta = 0.25</li> <li>■ 4'h8 - Delta = 0.125</li> <li>■ 4'h9 - Delta = 0.0625</li> <li>■ 4'hA - Delta = 0.03125</li> <li>■ 4'hB - Delta = 0.015625</li> <li>■ 4'hC to 4'hF - Invalid</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.6.33 ZCalPUSearchGainTV

- **Description:** ZCalPUSearchGainTV: Search code end-gain control for pull-up calibration
- **Size:** 9 bits
- **Offset:** 0xb0000+0x31e
- **Exists:** Always



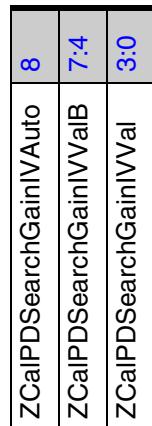
**Table 13-603 Fields for Register: ZCalPUSearchGainTV**

Bits	Name	Memory Access	Description
8	ZCalPUSearchGainTVAuto	R/W	<p>ZCalPUSearchGainTVAuto: Automatic gain change after first calibration. If asserted, the first pull-up calibration will end the code search using the gain specified in CSR ZCalPUSearchGainTVVal for the first calibration; and use the gain specified in CSR ZCalPUSearchGainTVValB for the second and subsequent calibrations</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>
7:4	ZCalPUSearchGainTVValB	R/W	<p>ZCalPUSearchGainTVValB: End-gain value to end the code search. This value is never used for the first calibration. It is used for second and subsequent calibrations when ZCalPUSearchGainTVAuto is 1. The gain value encoding is the same used for ZCalPUSearchGainTVVal</p> <p><b>Value After Reset:</b> 0x8</p> <p><b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
3:0	ZCalPUSearchGainTVVal	R/W	<p>ZCalPUSearchGainTVVal: End-gain value to end the code search. When the search algorithm get down to this gain value, as soon as the test result toggles, the search ends. This value is used for the first calibration and will be used for the subsequent ones only if ZCalPUSearchGainTVAuto is 0. The gain value encoding will result in the step deltas below:</p> <ul style="list-style-type: none"> <li>■ 4'h0 - Delta = 32.0</li> <li>■ 4'h1 - Delta = 16.0</li> <li>■ 4'h2 - Delta = 8.0</li> <li>■ 4'h3 - Delta = 4.0</li> <li>■ 4'h4 - Delta = 2.0</li> <li>■ 4'h5 - Delta = 1.0</li> <li>■ 4'h6 - Delta = 0.5</li> <li>■ 4'h7 - Delta = 0.25</li> <li>■ 4'h8 - Delta = 0.125</li> <li>■ 4'h9 - Delta = 0.0625</li> <li>■ 4'hA - Delta = 0.03125</li> <li>■ 4'hB - Delta = 0.015625</li> <li>■ 4'hC to 4'hF - Invalid</li> </ul> <p><b>Value After Reset:</b> 0x8</p> <p><b>Exists:</b> Always</p>

### 13.6.34 ZCalPDSearchGainIV

- **Description:** ZCalPDSearchGainIV: Search code initial-gain control for pull-down calibration
- **Size:** 9 bits
- **Offset:** 0xb0000+0x31f
- **Exists:** Always



**Table 13-604 Fields for Register: ZCalPDSearchGainIV**

Bits	Name	Memory Access	Description
8	ZCalPDSearchGainIVAuto	R/W	<p>ZCalPDSearchGainIVAuto: Automatic gain change after first calibration.</p> <p>If asserted, the first pull-down calibration will start the code search using the gain specified in CSR ZCalPDSearchGainIVVal for the first calibration; and use the gain specified in CSR ZCalPDSearchGainIVValB for the second and subsequent calibrations.</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>
7:4	ZCalPDSearchGainIVValB	R/W	<p>ZCalPDSearchGainIVValB: Initial-gain value to start code search with. This value is never used for the first calibration. It is used for the second and subsequent calibrations when ZCalPDSearchGainIVAuto is 1. The gain value encoding is the same used for ZCalPDSearchGainIVVal.</p> <p><b>Value After Reset:</b> 0x3</p> <p><b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
3:0	ZCalPDSearchGainIVVal	R/W	<p>ZCalPDSearchGainIVVal: Initial-gain value to start code search with. This value is used for the first calibration and will be used for the subsequent ones only if ZCalPDSearchGainIVAuto is 0. The gain value encoding will result in the step deltas below:</p> <ul style="list-style-type: none"> <li>■ 4'h0 - Delta = 32.0</li> <li>■ 4'h1 - Delta = 16.0</li> <li>■ 4'h2 - Delta = 8.0</li> <li>■ 4'h3 - Delta = 4.0</li> <li>■ 4'h4 - Delta = 2.0</li> <li>■ 4'h5 - Delta = 1.0</li> <li>■ 4'h6 - Delta = 0.5</li> <li>■ 4'h7 - Delta = 0.25</li> <li>■ 4'h8 - Delta = 0.125</li> <li>■ 4'h9 - Delta = 0.0625</li> <li>■ 4'hA - Delta = 0.03125</li> <li>■ 4'hB - Delta = 0.015625</li> <li>■ 4'hC to 4'hF - Invalid</li> </ul> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

### 13.6.35 ZQUpdate

- **Description:** ZQUpdate: Impedance Update
- **Size:** 1 bit
- **Offset:** 0xb0000+0x320
- **Exists:** Always



**Table 13-605 Fields for Register: ZQUpdate**

Bits	Name	Memory Access	Description
0	ZQUpdate	R/W	<p>ZQUpdate: A transition from 0 to 1 in this CSR will update the calibration codes going out to the IO slices, with the most recent calibrated codes. Refer to ZQCalCodePU and ZQCalCodePU descriptions.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.6.36 ZQCalCodePUchX (for X = 0; X <= 1)

- **Description:** ZQCalCodePUchX: Per channel Pull-up impedance calibration code sent to all the IOs
- **Size:** 9 bits
- **Offset:** 0xb0000+0x321+(X\*0xc)
- **Exists:** Always



**Table 13-606 Fields for Register: ZQCalCodePUchX (for X = 0; X <= 1)**

Bits	Name	Memory Access	Description
8:0	ZQCalCodePUch0	R/W	<p>ZQCalCodePUch0: Per channel Pull-up impedance calibration code sent to all the IOs Pull-up impedance calibration code sent to all the IOs Max and Min value for LP5 255 &amp; 0.</p> <p><b>Value After Reset:</b> 0x180</p> <p><b>Exists:</b> Always</p>

### 13.6.37 ZQCalCodePDchX (for X = 0; X <= 1)

- **Description:** ZQCalCodePDchX: channel 0 Pull-down impedance calibration code sent to all the IOs
- **Size:** 9 bits
- **Offset:** 0xb0000+0x322+(X\*0xc)
- **Exists:** Always



**Table 13-607 Fields for Register: ZQCalCodePDchX (for X = 0; X <= 1)**

Bits	Name	Memory Access	Description
8:0	ZQCalCodePDch0	R/W	<p>ZQCalCodePDch0: Per channel Pull-down impedance calibration code sent to all the IOs</p> <p><b>Value After Reset:</b> 0x180</p> <p><b>Exists:</b> Always</p>

### 13.6.38 ZQCalBaseCtrl

- **Name:** IOs Calibration Base legs control
- **Description:** ZQCalBaseCtrl: Deprecated
- **Size:** 2 bits
- **Offset:** 0xb0000+0x323
- **Exists:** Always



**Table 13-608 Fields for Register: ZQCalBaseCtrl**

Bits	Name	Memory Access	Description
1	ZQCalBasePD	R/W	ZQCalBasePD: Deprecated. Must be 0. <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always
0	ZQCalBasePU	R/W	ZQCalBasePU: Deprecated. Must be 0. <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always

### 13.6.39 ZQCalCodeOffsetPU

- **Name:** Pull-up impedance calibration code offset control
- **Description:** ZQCalCodeOffsetPU: This register enables the adjustment of the calibration code coming from the calibration circuit. An Offset value can be added or subtracted from ZCalCodePU before going to ZQCalCodePU CSR. It is not recommended to change this value from the default zero value.
- **Size:** 4 bits
- **Offset:** 0xb0000+0x324
- **Exists:** Always



**Table 13-609 Fields for Register: ZQCalCodeOffsetPU**

Bits	Name	Memory Access	Description
3:0	ZQCalCodeOffsetValPU	R/W	<p>ZQCalCodeOffsetValPU: Offset value 2's complement coded, to specify positive or negative offset values</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.6.40 ZQCalCodeOffsetPD

- **Name:** Pull-down impedance calibration code offset control
- **Description:** ZQCalCodeOffsetPD: This register enables the adjustment of the calibration code coming from the calibration circuit. An Offset value can be added or subtracted from ZCalCodePD before going to ZQCalCodePD CSR. It is not recommended to change this value from the default zero value.
- **Size:** 4 bits
- **Offset:** 0xb0000+0x325
- **Exists:** Always



**Table 13-610 Fields for Register: ZQCalCodeOffsetPD**

Bits	Name	Memory Access	Description
3:0	ZQCalCodeOffsetValPD	R/W	<p>ZQCalCodeOffsetValPD: Offset value 2's complement coded, to specify positive or negative offset values</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.6.41 ZQCalCodeOvrPU

- **Name:** Pull-up impedance Calibration code override
- **Description:** ZQCalCodeOvrPU: This CSR allow the pull-up impedance calibration code going to the IOs to be overridden. This CSR may only be written when the calibrator is not running.
- **Size:** 16 bits
- **Offset:** 0xb0000+0x326
- **Exists:** Always

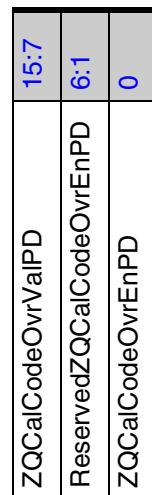


**Table 13-611 Fields for Register: ZQCalCodeOvrPU**

Bits	Name	Memory Access	Description
15:7	ZQCalCodeOvrValPU	R/W	<p>ZQCalCodeOvrValPU: If ZQCalCodeOvrEnPU is set, then the value provided here by software will be used as the pull-up calibration code to be propagated to the IOs, instead of the value obtained from the pull-up calibration. This value should be set only during initialization and 0 to 1 transition of ZQCalCodeOvrEnPU will transmit code.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
6:1	ReservedZQCalCodeOvrEnPU	R/W	<p>ReservedZQCalCodeOvrEnPU: Reserved for future use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
0	ZQCalCodeOvrEnPU	R/W	<p>ZQCalCodeOvrEnPU: 1'b1 - Enable the override of the pull-up calibration code</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.6.42 ZQCalCodeOvrPD

- **Name:** Pull-down impedance Calibration code override
- **Description:** ZQCalCodeOvrPD:  
This CSR allow the pull-down impedance calibration code going to the IOs to be overridden. This CSR may only be written when the calibrator is not running.
- **Size:** 16 bits
- **Offset:** 0xb0000+0x327
- **Exists:** Always



**Table 13-612 Fields for Register: ZQCalCodeOvrPD**

Bits	Name	Memory Access	Description
15:7	ZQCalCodeOvrValPD	R/W	<p>ZQCalCodeOvrValPD: If ZQCalCodeOvrEnPD is set, then the value provided here by software will be used as the pull-down calibration code to be propagated to the IOs, instead of the value obtained from the pull-down calibration. This value should be set only during initialization and 0 to 1 transition of ZQCalCodeOvrEnPD will transmit code.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
6:1	ReservedZQCalCodeOvrEnPD	R/W	<p>ReservedZQCalCodeOvrEnPD: Reserved for future use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
0	ZQCalCodeOvrEnPD	R/W	<p>ZQCalCodeOvrEnPD: 1'b1 - Enable the override of the pull-down calibration code.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.6.43 ZQCalCodePUMax

- **Description:** ZQCalCodePUMax: Max ZQCalCodePU value for overvoltage protection
- **Size:** 10 bits
- **Offset:** 0xb0000+0x328
- **Exists:** Always



**Table 13-613 Fields for Register: ZQCalCodePUMax**

Bits	Name	Memory Access	Description
9:0	ZQCalCodePUMax	R/W	<p>ZQCalCodePUMax: Max Pull up calibrated codes value that can be transmitted to IOs and PU drivers Max value for LP54X is 511</p> <p><b>Value After Reset:</b> 0x1ff</p> <p><b>Exists:</b> Always</p>

### 13.6.44 ZQCalCodePUMin

- **Description:** ZQCalCodePUMin: Min ZQCalCodePU value for overvoltage protection
- **Size:** 10 bits
- **Offset:** 0xb0000+0x329
- **Exists:** Always



**Table 13-614 Fields for Register: ZQCalCodePUMin**

Bits	Name	Memory Access	Description
9:0	ZQCalCodePUMin	R/W	<p>ZQCalCodePUMin: Min Pull PU calibrated codes value that can be transmitted to IOs and PU drivers Min value for LP54X is 256</p> <p><b>Value After Reset:</b> 0x100</p> <p><b>Exists:</b> Always</p>

### 13.6.45 ZQCalCodePDMax

- **Description:** ZQCalCodePDMax: Max ZQCalCodePD value for overvoltage protection
- **Size:** 10 bits
- **Offset:** 0xb0000+0x32a
- **Exists:** Always



**Table 13-615 Fields for Register: ZQCalCodePDMax**

Bits	Name	Memory Access	Description
9:0	ZQCalCodePDMax	R/W	<p>ZQCalCodePDMax: Max Pull PD calibrated codes value that can be transmitted to IOs and PD drivers For LP54X max value is 511</p> <p><b>Value After Reset:</b> 0x1ff</p> <p><b>Exists:</b> Always</p>

### 13.6.46 ZQCalCodePDMIn

- **Description:** ZQCalCodePDMIn: Min ZQCalCodePD value for overvoltage protection
- **Size:** 10 bits
- **Offset:** 0xb0000+0x32b
- **Exists:** Always



**Table 13-616 Fields for Register: ZQCalCodePDMIn**

Bits	Name	Memory Access	Description
9:0	ZQCalCodePDMIn	R/W	<p>ZQCalCodePDMIn: Min Pull PD calibrated codes value that can be transmitted to IOs and PD drivers Min value for LP54X is 256</p> <p><b>Value After Reset:</b> 0x100</p> <p><b>Exists:</b> Always</p>

### 13.6.47 ZCalStopClk\_pX (for X = 0; X <= 3)

- **Description:** ZCalStopClk\_pX: pstateable impedance calibration trigger to gate dficlk
- **Size:** 1 bit
- **Offset:** 0xb0000+0x32f+(X\*0x100000)
- **Exists:** Always



**Table 13-617 Fields for Register: ZCalStopClk\_pX (for X = 0; X <= 3)**

Bits	Name	Memory Access	Description
0	ZCalStopClk_p0	R/W	<p>ZCalStopClk_p0: This CSR is used to stop dficlk when zcal is not in use. In other words, it acts as an active low enable to the clock gater circuit. When this csr is pulled to low, it gates Dficlk to Zcal HM and PUB Zcal If ZCalStopClk=1, the calibrator will not run. ZcalStopCLK should be set to low during active calibration. When ZCalStopClk = 1, it turns off the cfgclk too, this implies that the CSR slaves, Voltage regulators and Analog-test Bus functionality are un-useable. This csr will not do anything while Reset is asserted.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

## 13.7 **DWC\_DDRPHYA\_DRTUBj Registers**

### 13.7.1 PieVecCfg

- **Name:** Start vector value to be used for LP3-exit or Init PIE Sequence
- **Description:** PieVecCfg; Programmed by dwc\_ddrphy\_physinit\_I\_loadPIEImage() to support frequency changes. Defines the start vector to be used in LP3-exit or Init PIE sequence.
- **Size:** 16 bits
- **Offset:** 0xc0000+0x0
- **Exists:** Always

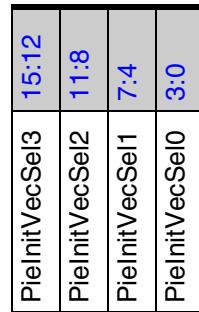


**Table 13-618 Fields for Register: PieVecCfg**

Bits	Name	Memory Access	Description
15:12	PielnitStartVec3	R/W	PielnitStartVec3: PIE start vector3 used on Initialization/Retention Exit. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
11:8	PielnitStartVec2	R/W	PielnitStartVec2: PIE start vector2 used on Initialization/Retention Exit. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7:4	PielnitStartVec1	R/W	PielnitStartVec1: PIE start vector1 used on Initialization/Retention Exit. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
3:0	PielnitStartVec0	R/W	PielnitStartVec0: PIE start vector0 used on Initialization/Retention Exit. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.7.2 PieInitVecSel

- **Name:** Start vector value to be used for LP3-exit or Init PIE Sequence
- **Description:** PieInitVecSel: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes. Set the start vector to be used in LP3-exit or Init PIE sequence.
- **Size:** 16 bits
- **Offset:** 0xc0000+0x1
- **Exists:** Always



**Table 13-619 Fields for Register: PieInitVecSel**

Bits	Name	Memory Access	Description
15:12	PieInitVecSel3	R/W	PieInitVecSel3: Programmable Initialization Vector Select. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
11:8	PieInitVecSel2	R/W	PieInitVecSel2: Programmable Initialization Vector Select. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7:4	PieInitVecSel1	R/W	PieInitVecSel1: Programmable Initialization Vector Select. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
3:0	PieInitVecSel0	R/W	PieInitVecSel0: Programmable Initialization Vector Select. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.7.3 DctShadowRegs

- **Description:** DctShadowRegs: Reserved for PHY training firmware use.
- **Size:** 1 bit
- **Offset:** 0xc0000+0x4
- **Exists:** Always



**Table 13-620 Fields for Register: DctShadowRegs**

Bits	Name	Memory Access	Description
0	DctShadowRegs	R	<p>DctShadowRegs: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.7.4 DctWriteOnlyShadow

- **Description:** DctWriteOnlyShadow: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** 0xc0000+0x30
- **Exists:** Always



**Table 13-621 Fields for Register: DctWriteOnlyShadow**

Bits	Name	Memory Access	Description
15:0	DctWriteOnlyShadow	R	<p>DctWriteOnlyShadow: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.7.5 UctWriteOnly

- **Description:** UctWriteOnly: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** 0xc0000+0x32
- **Exists:** Always



**Table 13-622 Fields for Register: UctWriteOnly**

Bits	Name	Memory Access	Description
15:0	UctWriteOnly	R/W	<p>UctWriteOnly: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.7.6 UctWriteProt

- **Description:** UctWriteProt: Reserved for PHY training firmware use.
- **Size:** 1 bit
- **Offset:** 0xc0000+0x33
- **Exists:** Always



**Table 13-623 Fields for Register: UctWriteProt**

Bits	Name	Memory Access	Description
0	UctWriteProt	R/W	<p>UctWriteProt: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

### 13.7.7 UctDatWriteOnly

- **Description:** UctDatWriteOnly: Reserved for PHY training firmware use.
- **Size:** 16 bits
- **Offset:** 0xc0000+0x34
- **Exists:** Always



**Table 13-624 Fields for Register: UctDatWriteOnly**

Bits	Name	Memory Access	Description
15:0	UctDatWriteOnly	R/W	<p>UctDatWriteOnly: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.7.8 UctDatWriteProt

- **Description:** UctDatWriteProt: Reserved for PHY training firmware use.
- **Size:** 1 bit
- **Offset:** 0xc0000+0x35
- **Exists:** Always



**Table 13-625 Fields for Register: UctDatWriteProt**

Bits	Name	Memory Access	Description
0	UctDatWriteProt	R/W	UctDatWriteProt: Reserved for PHY training firmware use. <b>Value After Reset:</b> 0x1 <b>Exists:</b> Always

### 13.7.9 UctlErr

- **Description:** UctlErr: Reserved for PHY training firmware use.
- **Size:** 1 bit
- **Offset:** 0xc0000+0x36
- **Exists:** Always



**Table 13-626 Fields for Register: UctlErr**

Bits	Name	Memory Access	Description
0	UctlErr	R/W	<p>UctlErr: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.7.10 DRTUBParityInvert

- **Description:** DRTUBParityInvert: Invert APB Parity for register slave DRTUB
- **Size:** 2 bits
- **Offset:** 0xc0000+0x4d
- **Exists:** Always



**Table 13-627 Fields for Register: DRTUBParityInvert**

Bits	Name	Memory Access	Description
1:0	DRTUBParityInvert	R/W	<p>DRTUBParityInvert: Invert APB Parity for register slave DRTUB. As required for Automotive. NOTE: This register should be used only for test. Set the bits for only one slave at a time. When bits are set for a particular slave. APB Reads of only that slave are valid. Bit 0 applies to [7:0] Bit 1 applies to [15:8] In HMZCAL, When ZcalStopClk=1, this csr can not be accessed.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.7.11 UCParityInvert

- **Description:** UCParityInvert: Invert APB Parity for ARC \*CCM Reads
- **Size:** 4 bits
- **Offset:** 0xc0000+0x4e
- **Exists:** Always



**Table 13-628 Fields for Register: UCParityInvert**

Bits	Name	Memory Access	Description
3:0	UCParityInvert	R/W	<p>UCParityInvert: Invert APB Parity for ARC *CCM Reads. As required for Automotive. Bit 0 applies to [7:0] Bit 1 applies to [15:8] Bit 2 applies to [23:16] Bit 3 applies to [31:24]</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.7.12 ScratchPadDRTUB

- **Description:** ScratchPadDRTUB: ScratchPad for DRTUB
- **Size:** 16 bits
- **Offset:** 0xc0000+0x7d
- **Exists:** Always



**Table 13-629 Fields for Register: ScratchPadDRTUB**

Bits	Name	Memory Access	Description
15:0	ScratchPadDRTUB	R/W	<p>ScratchPadDRTUB: ScratchPad for DRTUB. As required for Automotive. In HMZCAL, When ZcalStopClk=1, this csr can not be accessed.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.7.13 UcclkHclkEnables

- **Name:** Ucclk and Hclk enables
- **Description:** UcclkHclkEnables: UcClk/Hclk Enables. [UcClk and Hclk run at either DFICLK or are gated]
- **Size:** 3 bits
- **Offset:** 0xc0000+0x80
- **Exists:** Always

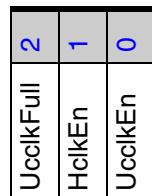


Table 13-630 Fields for Register: UcclkHclkEnables

Bits	Name	Memory Access	Description
2	UcclkFull	R/W	<p>UcclkFull: When set, allows the ARC to run at full DfiClk Speed. This bit must always be 1. A value of 0 is not supported.</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>
1	HclkEn	R/W	<p>HclkEn: Enables the clock to the ARC. When training has completed (and assuming no further need for the training hardware), the enable should be set to 0 to reduce power.</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>
0	UcclkEn	R/W	<p>UcclkEn: Enables the clock to ARC Memory. When training has completed (and assuming no further need for the microcontroller), the enable should be set to 0 to reduce power.</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

### 13.7.14 ArcEccIndications

- **Name:** Indications of ARC double-bit ECC errors
- **Description:** ArcEccIndications: Indications of ARC double bit errors, caused either by ARC reads of \*CCM or by APB-sourced reads of \*CCM Indications are cleared by resetting the ARC If any bits are set, a the PhyEccErr interrupt is posted.
- **Size:** 2 bits
- **Offset:** 0xc0000+0x82
- **Exists:** Always



Table 13-631 Fields for Register: ArcEccIndications

Bits	Name	Memory Access	Description
1	ArclccmDbError	R	<p>ArclccmDbError: Double-bit error in the ICCM Data</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
0	ArcDccmDbError	R	<p>ArcDccmDbError: Double-bit error in the DCCM Data</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.7.15 ArcIccmSbErrCtr

- **Description:** ArcIccmSbErrCtr: Count of ICCM Single-bit corrected ECC Errors
- **Size:** 16 bits
- **Offset:** 0xc0000+0x83
- **Exists:** Always



**Table 13-632 Fields for Register: ArcIccmSbErrCtr**

Bits	Name	Memory Access	Description
15:0	ArcIccmSbErrCtr	R	<p>ArcIccmSbErrCtr: Count of ICCM Single-bit corrected ECC Errors generated from ARC-sourced transactions. Controlled by csrArcIccmSbCtrEn. Note that this also counts uncorrected errors detected in speculative preload. The counter freezes at its maximum value.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.7.16 ArcDccmSbErrCtr

- **Description:** ArcDccmSbErrCtr: Count of DCCM Single-bit corrected ECC Errors
- **Size:** 16 bits
- **Offset:** 0xc0000+0x84
- **Exists:** Always



**Table 13-633 Fields for Register: ArcDccmSbErrCtr**

Bits	Name	Memory Access	Description
15:0	ArcDccmSbErrCtr	R	<p>ArcDccmSbErrCtr: Count of DCCM Single-bit corrected ECC Errors generated from ARC-sourced transactions. Controlled by csrArcDccmSbCtrEn. The counter freezes at its maximum value.</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always  <b>Reset Mask:</b> 0x0  <b>Volatile:</b> true</p>

### 13.7.17 ArcSbCtrEnables

- **Name:** Enables for single-bit error counters
- **Description:** ArcSbCtrEnables: Setting the bit enables the counter. Clearing the bit stops and clears the counter.
- **Size:** 2 bits
- **Offset:** 0xc0000+0x85
- **Exists:** Always

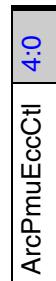


**Table 13-634 Fields for Register: ArcSbCtrEnables**

Bits	Name	Memory Access	Description
1	ArcDccmSbCtrEn	R/W	ArcDccmSbCtrEn: Enable for counter of single-bit data errors in the DCCM Data <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
0	ArcIccmSbCtrEn	R/W	ArcIccmSbCtrEn: Enable for counter of single-bit data errors in the ICCM Data <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.7.18 ArcPmuEccCtl

- **Description:** ArcPmuEccCtl: ARC ECC Control
- **Size:** 5 bits
- **Offset:** 0xc0000+0x86
- **Exists:** Always



**Table 13-635 Fields for Register: ArcPmuEccCtl**

Bits	Name	Memory Access	Description
4:0	ArcPmuEccCtl	R/W	<p>ArcPmuEccCtl: Overrides/Control for ARC Error Protection Hardware Control Register (ERP_CTRL) Modify this register only during initialization before any firmware is loaded. [4] engages override. When this bit is set, [3:0] have the functions listed below [3] DCCM exception disable. [2] ICCM exception disable. [1] DCCM ECC disable. [0] ICCM ECC Disable</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.7.19 StartDCCMClear

- **Description:** StartDCCMClear: Enables the clearing of the DCCM by the PHY
- **Size:** 1 bit
- **Offset:** 0xc0000+0x88
- **Exists:** Always



**Table 13-636 Fields for Register: StartDCCMClear**

Bits	Name	Memory Access	Description
0	StartDCCMClear	R/W	<p>StartDCCMClear: A 0-&gt;1 transition on this bit starts a clear of the DCCM. The ARC should be disabled when this function is operating. While the function is operating, APB writes to ICCM or DCCM will appear to complete but they will be ignored. APB accesses to internal PHY registers can occur while this function is running. This operation takes less than 8200 DfiClk cycles when csrUcclkFull=1, and less than 16400 DfiClk cycles when csrUcclkFull=0. This bit must be written to 0 when the clear operation completes (after csrDCCMClearRunning transitions to 0).</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.7.20 DCCMClearRunning

- **Description:** DCCMClearRunning: Indicates when DCCM clearing is in progress.
- **Size:** 1 bit
- **Offset:** 0xc0000+0x89
- **Exists:** Always



**Table 13-637 Fields for Register: DCCMClearRunning**

Bits	Name	Memory Access	Description
0	DCCMClearRunning	R	<p>DCCMClearRunning: When 1, indicates that the clearing of the DCCM by the PHY is in progress.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.7.21 PIEMicroReset

- **Name:** Controls reset and clock shutdown on the local microcontroller by the PIE.
- **Description:** PIEMicroReset: When enabled by csrMicroResetPIEEn, controls reset and clock shutdown on the local microcontroller.
- **Size:** 4 bits
- **Offset:** 0xc0000+0x99
- **Exists:** Always



**Table 13-638 Fields for Register: PIEMicroReset**

Bits	Name	Memory Access	Description
3	PIEResetToMicro	R/W	<p>PIEResetToMicro: This is an analogue to csrResetToMicro. This register is set by the PIE and is enabled by MicroResetPIEEn. The PIE can set this bit to hold the microcontroller in reset by hardware. Typically, this bit is used at power up to hold the program counter at the boot vector while BIOS loads the microcontroller program code. While stalled, the microcontroller clocks are gated off for power reduction. This reset initializes the program counter to begin execution from the boot vector. This reset also clears the interrupt sticky bits.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>
2:1	PIEMicroResetReserved	R/W	<p>PIEMicroResetReserved: Reserved.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
0	PIESTallToMicro	R/W	<p>PIESTallToMicro: This is an analogue to csrStallToMicro. This register is set by the PIE and is enabled by MicroResetPIEEn. The PIE can set this bit to stall the microcontroller by hardware. After it is set, the UC is stalled after the delay specified in csrPIEMicroStallDelay. Typically, this bit is used at power up to hold the program counter at the boot vector while BIOS loads the microcontroller program code. While stalled, the microcontroller clocks are gated off for power reduction. This bit is forced to 0 on a rising edge of csrClearPIESTallToMicro.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.7.22 CUSTPHYREV

- **Description:** CUSTPHYREV: Customer settable by the customer
- **Size:** 6 bits
- **Offset:** 0xc0000+0xec
- **Exists:** Always



**Table 13-639 Fields for Register: CUSTPHYREV**

Bits	Name	Memory Access	Description
5:0	CUSTPHYREV	R	<p>CUSTPHYREV: The customer settable PHY version number. The value is derived from the `define DWC_LPDDR5XPHY_CUST_PHYREV</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.7.23 CUSTPUBREV

- **Description:** CUSTPUBREV: Customer settable by the customer
- **Size:** 6 bits
- **Offset:** 0xc0000+0xed
- **Exists:** Always

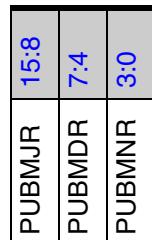


**Table 13-640 Fields for Register: CUSTPUBREV**

Bits	Name	Memory Access	Description
5:0	CUSTPUBREV	R	<p>CUSTPUBREV: The customer settable PUB version number. The value is derived from the `define DWC_LPDDR5XPHY_CUST_PUBREV</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.7.24 PUBREV

- **Description:** PUBREV: The hardware version of this PUB, excluding the PHY
- **Size:** 16 bits
- **Offset:** 0xc0000+0xee
- **Exists:** Always

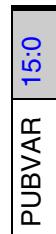


**Table 13-641 Fields for Register: PUBREV**

Bits	Name	Memory Access	Description
15:8	PUBMJR	R	<p>PUBMJR: Indicates major revision of the PUB.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
7:4	PUBMDR	R	<p>PUBMDR: Indicates moderate revision of the PUB.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>
3:0	PUBMNR	R	<p>PUBMNR: Indicates minor update of the PUB.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.7.25 PUBVAR

- **Description:** PUBVAR: The hardware variant of this PUB
- **Size:** 16 bits
- **Offset:** 0xc0000+0xef
- **Exists:** Always

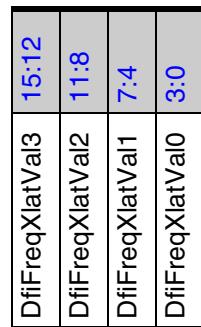


**Table 13-642 Fields for Register: PUBVAR**

Bits	Name	Memory Access	Description
15:0	PUBVAR	R	<p>PUBVAR: The hardware variant of this PUB 0xFFFF indicates the mainline release. Variants are identified by specific hex value.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.7.26 DfiFreqXlatX (for X = 0; X <= 15)

- **Name:** DFI Frequency Translation Register 0
- **Description:** DfiFreqXlatX: DFI Frequency Translation Register 0 Programmed by dwc\_ddrphy\_phy-init\_C\_initPhyConfig() to support frequency changes.
- **Size:** 16 bits
- **Offset:** 0xc0000+0xf0+(X\*0x1)
- **Exists:** Always



**Table 13-643 Fields for Register: DfiFreqXlatX (for X = 0; X <= 15)**

Bits	Name	Memory Access	Description
15:12	DfiFreqXlatVal3	R/W	DfiFreqXlatVal3: The sequencer start vector used when dfi_freq value is 3. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
11:8	DfiFreqXlatVal2	R/W	DfiFreqXlatVal2: The sequencer start vector used when dfi_freq value is 2. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
7:4	DfiFreqXlatVal1	R/W	DfiFreqXlatVal1: The sequencer start vector used when dfi_freq value is 1. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always
3:0	DfiFreqXlatVal0	R/W	DfiFreqXlatVal0: The sequencer start vector used when dfi_freq value is 0. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

## 13.8 **DWC\_DDRPHYA\_APBONLYj Registers**

### 13.8.1 MicroContMuxSel

- **Description:** MicroContMuxSel: PMU Config Mux Select
- **Size:** 1 bit
- **Offset:** 0xd0000+0x0
- **Exists:** Always



**Table 13-644 Fields for Register: MicroContMuxSel**

Bits	Name	Memory Access	Description
0	MicroContMuxSel	R/W	<p>MicroContMuxSel: This register controls access to the PHY configuration registers.</p> <ul style="list-style-type: none"> <li>■ 1 = MicroController/PIE (ARC) has control of csr bus. APB can only access APB_ONLY slave CSR's.</li> <li>■ 0 = Allow APB Access to csr bus. MicroController (ARC)/PIE csr requests are not blocked.</li> </ul> <p><b>Note:</b> When writing host Firmware that interacts with the ARC processor using the message interface, and receiving a message from the ARC that it is time to clear csrMicroContMuxSel, Firmware should wait 40 DfiClk before writing the bit to allow the ARC to finish message processing. The PIE can read and write CSRs regardless of how this bit is set.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.8.2 ContextToMicro

- **Description:** ContextToMicro: Determines which set of CSRs the ARC has access to
- **Size:** 1 bit
- **Offset:** 0xd0000+0x1
- **Exists:** Always



**Table 13-645 Fields for Register: ContextToMicro**

Bits	Name	Memory Access	Description
0	ContextToMicro	R/W	<p>ContextToMicro: Determines which set of CSRs the ARC has access to</p> <ul style="list-style-type: none"> <li>■ 1 = ARC will access external slave CSRs via AHB-Lite bus</li> <li>■ 0 = ARC will access internal PHY CSRs via Config Bus</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.8.3 ExternalAHBReset

- **Description:** ExternalAHBReset: Directly connected to Reset pin of external AHB-Lite interface
- **Size:** 1 bit
- **Offset:** 0xd0000+0x2
- **Exists:** Always



**Table 13-646 Fields for Register: ExternalAHBReset**

Bits	Name	Memory Access	Description
0	ExternalAHBReset	R/W	<p>ExternalAHBReset: Directly connected to Reset pin of external AHB-Lite interface</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.8.4 TDRDisable

- **Description:** TDRDisable: Disables TDR Interface
- **Size:** 1 bit
- **Offset:** 0xd0000+0x3
- **Exists:** Always



**Table 13-647 Fields for Register: TDRDisable**

Bits	Name	Memory Access	Description
0	TDRDisable	R/W	<p>TDRDisable: When set, disables JTAG/TDR Interface</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.8.5 UctShadowRegs

- **Name:** PMU/Controller Protocol - Controller Read-only Shadow
- **Description:** UctShadowRegs: This is used for the mailbox protocol between the firmware and the system. See section 3.4.1 of the training firmware AppNote for details on the protocol.
- **Size:** 2 bits
- **Offset:** 0xd0000+0x4
- **Exists:** Always



**Table 13-648 Fields for Register: UctShadowRegs**

Bits	Name	Memory Access	Description
1	UctDatWriteProtShadow	R	<p>UctDatWriteProtShadow: Reserved for future use.  <b>Value After Reset:</b> 0x0  <b>Exists:</b> Always  <b>Reset Mask:</b> 0x0  <b>Volatile:</b> true</p>
0	UctWriteProtShadow	R	<p>UctWriteProtShadow: When set to 0, the PMU has a message for the user  <b>Value After Reset:</b> 0x0  <b>Exists:</b> Always  <b>Reset Mask:</b> 0x0  <b>Volatile:</b> true</p>

### 13.8.6 BlockDfiShadowRegs

- **Name:** BlockDfiInterface - Read-only Shadow
- **Description:** BlockDfiShadowRegs: Monitors operations when PState info is being transferred from DCCM
- **Size:** 2 bits
- **Offset:** 0xd0000+0x5
- **Exists:** Always

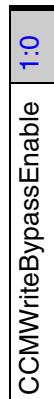


**Table 13-649 Fields for Register: BlockDfiShadowRegs**

Bits	Name	Memory Access	Description
1	PmuBusyShadow	R	<b>PmuBusyShadow:</b> Read-only copy of PmuBusy <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always <b>Reset Mask:</b> 0x0 <b>Volatile:</b> true
0	BlockDfiInterfaceEnShadow	R	<b>BlockDfiInterfaceEnShadow:</b> Read-only copy of BlockDfiInterfaceEn <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always <b>Reset Mask:</b> 0x0 <b>Volatile:</b> true

### 13.8.7 CCMWriteBypassEnable

- **Description:** CCMWriteBypassEnable: ARC Write Bypass Enable
- **Size:** 2 bits
- **Offset:** 0xd0000+0x8
- **Exists:** Always



**Table 13-650 Fields for Register: CCMWriteBypassEnable**

Bits	Name	Memory Access	Description
1:0	CCMWriteBypassEnable	R/W	<p>CCMWriteBypassEnable: Provides faster APB Write access to ICCM and DCCM memories When enabled, always write data in aligned 8 byte chunks, ARC must not be enabled (i.e., an even/odd pair of 32b words)</p> <ul style="list-style-type: none"> <li>■ 0X = Enabled when csrHclkEn=0</li> <li>■ 10 = Always Enabled</li> <li>■ 11 = Always Disabled</li> </ul> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.8.8 DctWriteOnly

- **Description:** DctWriteOnly: Reserved for future use.
- **Size:** 16 bits
- **Offset:** 0xd0000+0x30
- **Exists:** Always



**Table 13-651 Fields for Register: DctWriteOnly**

Bits	Name	Memory Access	Description
15:0	DctWriteOnly	R/W	DctWriteOnly: Reserved for future use. <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.8.9 DctWriteProt

- **Description:** DctWriteProt: DCT downstream mailbox protocol CSR.
- **Size:** 1 bit
- **Offset:** 0xd0000+0x31
- **Exists:** Always



**Table 13-652 Fields for Register: DctWriteProt**

Bits	Name	Memory Access	Description
0	DctWriteProt	R/W	<p>DctWriteProt: By setting this register to 0, the user acknowledges the receipt of the message.</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

### 13.8.10 UctWriteOnlyShadow

- **Description:** UctWriteOnlyShadow: Read-only view of the csr UctDatWriteOnly
- **Size:** 16 bits
- **Offset:** 0xd0000+0x32
- **Exists:** Always



**Table 13-653 Fields for Register: UctWriteOnlyShadow**

Bits	Name	Memory Access	Description
15:0	UctWriteOnlyShadow	R	<p>UctWriteOnlyShadow: Used to pass the message ID for major messages. Also used to pass the lower 16 bits for streaming messages.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.8.11 UctDatWriteOnlyShadow

- **Description:** UctDatWriteOnlyShadow: Read-only view of the csr UctDatWriteOnly
- **Size:** 16 bits
- **Offset:** 0xd0000+0x34
- **Exists:** Always

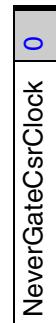


**Table 13-654 Fields for Register: UctDatWriteOnlyShadow**

Bits	Name	Memory Access	Description
15:0	UctDatWriteOnlyShadow	R	<p>UctDatWriteOnlyShadow: Used to pass the upper 16 bits for streaming messages. Not used in passing major messages.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.8.12 NeverGateCsrClock

- **Description:** NeverGateCsrClock: Reserved for PHY training firmware use.
- **Size:** 1 bit
- **Offset:** 0xd0000+0x35
- **Exists:** Always

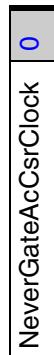


**Table 13-655 Fields for Register: NeverGateCsrClock**

Bits	Name	Memory Access	Description
0	NeverGateCsrClock	R/W	<p>NeverGateCsrClock: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.8.13 NeverGateAcCsrClock

- **Description:** NeverGateAcCsrClock: Reserved for PHY training firmware use.
- **Size:** 1 bit
- **Offset:** 0xd0000+0x36
- **Exists:** Always

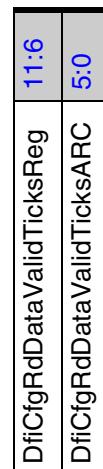


**Table 13-656 Fields for Register: NeverGateAcCsrClock**

Bits	Name	Memory Access	Description
0	NeverGateAcCsrClock	R/W	<p>NeverGateAcCsrClock: Reserved for PHY training firmware use.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.8.14 DfiCfgRdDataValidTicks

- **Name:** Bounding number of DfiClk ticks required for valid csr Rd Data on APB reads.
- **Description:** DfiCfgRdDataValidTicks: Roundtrip delay of a register read access via APB. There are separate CSRs for accesses via PIE and UC. This value must exceed the max number of required ticks, and not be changed from its reset value. The actual read may take less time
- **Size:** 12 bits
- **Offset:** 0xd0000+0x37
- **Exists:** Always



**Table 13-657 Fields for Register: DfiCfgRdDataValidTicks**

Bits	Name	Memory Access	Description
11:6	DfiCfgRdDataValidTicksReg	R/W	<p>DfiCfgRdDataValidTicksReg: Value for APB reads of everything except ICCM and DCCM</p> <p><b>Value After Reset:</b> 0x10</p> <p><b>Exists:</b> Always</p>
5:0	DfiCfgRdDataValidTicksARC	R/W	<p>DfiCfgRdDataValidTicksARC: Value for APB reads of ICCM and DCCM</p> <p><b>Value After Reset:</b> 0x18</p> <p><b>Exists:</b> Always</p>

### 13.8.15 DisableHMRdSpeedUp

- **Description:** DisableHMRdSpeedUp: Deprecated.
- **Size:** 1 bit
- **Offset:** 0xd0000+0x39
- **Exists:** Always



**Table 13-658 Fields for Register: DisableHMRdSpeedUp**

Bits	Name	Memory Access	Description
0	DisableHMRdSpeedUp	R/W	<p>DisableHMRdSpeedUp: Deprecated</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.8.16 OverrideHMRdSpeedUp

- **Description:** OverrideHMRdSpeedUp: Override for logic that disables CSR Read Speedup for older Hard Macros
- **Size:** 2 bits
- **Offset:** 0xd0000+0x3a
- **Exists:** Always



**Table 13-659 Fields for Register: OverrideHMRdSpeedUp**

Bits	Name	Memory Access	Description
1:0	OverrideHMRdSpeedUp	R/W	<p>OverrideHMRdSpeedUp: This register should be used only for Debug. 00: Disabled 10: Configured for Older Hard Macros X1: Configured for newer Hard Macros</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.8.17 Dfi0DebugControl

- **Description:** Dfi0DebugControl: Reserved for PHY training firmware use and for debugging
- **Size:** 16 bits
- **Offset:** 0xd0000+0x40
- **Exists:** Always



**Table 13-660 Fields for Register: Dfi0DebugControl**

Bits	Name	Memory Access	Description
15:0	Dfi0DebugControl	R/W	<p>Dfi0DebugControl: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.8.18 Dfi0DebugCaptureX (for X = 0; X <= 1)

- **Description:** Dfi0DebugCaptureX: Reserved for PHY training firmware use and for debugging
- **Size:** 13 bits
- **Offset:** 0xd0000+0x41+(X\*0x1)
- **Exists:** Always



**Table 13-661 Fields for Register: Dfi0DebugCaptureX (for X = 0; X <= 1)**

Bits	Name	Memory Access	Description
12:0	Dfi0DebugCapture0	R	<p>Dfi0DebugCapture0: Reserved for PHY training firmware use and for debugging</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.8.19 Dfi0DebugPerfCtrEn

- **Description:** Dfi0DebugPerfCtrEn: Reserved for PHY training firmware use and for debugging
- **Size:** 1 bit
- **Offset:** 0xd0000+0x43
- **Exists:** Always



Table 13-662 Fields for Register: Dfi0DebugPerfCtrEn

Bits	Name	Memory Access	Description
0	Dfi0DebugPerfCtrEn	R/W	<p>Dfi0DebugPerfCtrEn: Reserved for PHY training firmware use and for debugging</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.8.20 Dfi0DebugPerfCtr

- **Description:** Dfi0DebugPerfCtr: Reserved for PHY training firmware use and for debugging
- **Size:** 16 bits
- **Offset:** 0xd0000+0x44
- **Exists:** Always



**Table 13-663 Fields for Register: Dfi0DebugPerfCtr**

Bits	Name	Memory Access	Description
15:0	Dfi0DebugPerfCtr	R	<p>Dfi0DebugPerfCtr: Reserved for PHY training firmware use and for debugging</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.8.21 Dfi1DebugControl

- **Description:** Dfi1DebugControl: Reserved for PHY training firmware use and for debugging
- **Size:** 16 bits
- **Offset:** 0xd0000+0x48
- **Exists:** Always



**Table 13-664 Fields for Register: Dfi1DebugControl**

Bits	Name	Memory Access	Description
15:0	Dfi1DebugControl	R/W	<p>Dfi1DebugControl: Reserved for Synopsys internal use</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.8.22 Dfi1DebugCaptureX (for X = 0; X <= 1)

- **Description:** Dfi1DebugCaptureX: Reserved for PHY training firmware use and for debugging
- **Size:** 13 bits
- **Offset:** 0xd0000+0x49+(X\*0x1)
- **Exists:** Always



**Table 13-665 Fields for Register: Dfi1DebugCaptureX (for X = 0; X <= 1)**

Bits	Name	Memory Access	Description
12:0	Dfi1DebugCapture0	R	<p>Dfi1DebugCapture0: Reserved for PHY training firmware use and for debugging</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.8.23 Dfi1DebugPerfCtrEn

- **Description:** Dfi1DebugPerfCtrEn: Reserved for PHY training firmware use and for debugging
- **Size:** 1 bit
- **Offset:** 0xd0000+0x4b
- **Exists:** Always



Table 13-666 Fields for Register: Dfi1DebugPerfCtrEn

Bits	Name	Memory Access	Description
0	Dfi1DebugPerfCtrEn	R/W	<p>Dfi1DebugPerfCtrEn: Reserved for PHY training firmware use and for debugging</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.8.24 Dfi1DebugPerfCtr

- **Description:** Dfi1DebugPerfCtr: Reserved for PHY training firmware use and for debugging
- **Size:** 16 bits
- **Offset:** 0xd0000+0x4c
- **Exists:** Always



**Table 13-667 Fields for Register: Dfi1DebugPerfCtr**

Bits	Name	Memory Access	Description
15:0	Dfi1DebugPerfCtr	R	<p>Dfi1DebugPerfCtr: Reserved for PHY training firmware use and for debugging</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.8.25 APBONLYParityInvert

- **Description:** APBONLYParityInvert: Invert APB Parity for register slave APBONLY
- **Size:** 2 bits
- **Offset:** 0xd0000+0x4d
- **Exists:** Always



**Table 13-668 Fields for Register: APBONLYParityInvert**

Bits	Name	Memory Access	Description
1:0	APBONLYParityInvert	R/W	<p>APBONLYParityInvert: Invert APB Parity for register slave APBONLY. As required for Automotive. NOTE: This register should be used only for test. Set the bits for only one slave at a time. When bits are set for a particular slave. APB Reads of only that slave are valid. Bit 0 applies to [7:0] Bit 1 applies to [15:8] In HMZCAL, When ZcalStopClk=1, this csr can not be accessed.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.8.26 WaitCondAPB

- **Description:** WaitCondAPB: Programmed by dwc\_ddrphy\_phyinit\_I\_loadPIEImage() to support frequency changes.
- **Size:** 1 bit
- **Offset:** 0xd0000+0x50
- **Exists:** Always



**Table 13-669 Fields for Register: WaitCondAPB**

Bits	Name	Memory Access	Description
0	WaitCondAPB	R/W	<p>WaitCondAPB: Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.8.27 PIERdDataValidTicks

- **Name:** Number of DfiClks between CfgBusRd and CfgBusDone on PIE reads
- **Description:** PIERdDataValidTicks: Tells the PIE HW how long to wait for valid data when issuing a CSR read Two DfiClks after this, the read data will appear in the dest GPR This CSR is for the PIE; see separate CSRs for reads via APB and UC. This value must not be changed from its reset value.
- **Size:** 12 bits
- **Offset:** 0xd0000+0x51
- **Exists:** Always

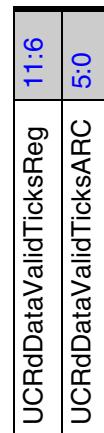
PIERdDataValidTicksReg	11:6
PIERdDataValidTicksARC	5:0

Table 13-670 Fields for Register: PIERdDataValidTicks

Bits	Name	Memory Access	Description
11:6	PIERdDataValidTicksReg	R/W	PIERdDataValidTicksReg: Value for PIE reads of everything except ICCM and DCCM This value must match the PIE_READ_GPR_VALID_TICKS value in hwt_seq0b_api.h <b>Value After Reset:</b> 0xe <b>Exists:</b> Always
5:0	PIERdDataValidTicksARC	R/W	PIERdDataValidTicksARC: Value for PIE reads of ICCM and DCCM <b>Value After Reset:</b> 0xc <b>Exists:</b> Always

### 13.8.28 UCRdDataValidTicks

- **Name:** Backup timeout counter for ARC register reads
- **Description:** UCRdDataValidTicks: Provides backup timeout counter for ARC register reads of unimplemented CSR slaves
- **Size:** 12 bits
- **Offset:** 0xd0000+0x52
- **Exists:** Always



**Table 13-671 Fields for Register: UCRdDataValidTicks**

Bits	Name	Memory Access	Description
11:6	UCRdDataValidTicksReg	R/W	UCRdDataValidTicksReg: This value must not be changed from its reset value. Timeout value for ARC register reads of unimplemented CSR slaves <b>Value After Reset:</b> 0xf <b>Exists:</b> Always
5:0	UCRdDataValidTicksARC	R/W	UCRdDataValidTicksARC: Reserved <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always

### 13.8.29 ScratchPadAPBONLY

- **Description:** ScratchPadAPBONLY: ScratchPad for APBONLY
- **Size:** 16 bits
- **Offset:** 0xd0000+0x7d
- **Exists:** Always



**Table 13-672 Fields for Register: ScratchPadAPBONLY**

Bits	Name	Memory Access	Description
15:0	ScratchPadAPBONLY	R/W	<p>ScratchPadAPBONLY: ScratchPad for APBONLY. As required for Automotive. In HMZCAL, When ZcalStopClk=1, this csr can not be accessed.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.8.30 MicroReset

- **Name:** Controls reset and clock shutdown on the local microcontroller
- **Description:** MicroReset: Controls reset and clock shutdown on the local microcontroller.
- **Size:** 4 bits
- **Offset:** 0xd0000+0x99
- **Exists:** Always



Table 13-673 Fields for Register: MicroReset

Bits	Name	Memory Access	Description
3	ResetToMicro	R/W	<p>ResetToMicro: This field is programmed as follows:</p> <ul style="list-style-type: none"> <li>■ Set this bit to apply synchronous reset to the microcontroller.</li> <li>■ Clear this bit to release reset to the microcontroller.</li> </ul> <p>This reset initializes the program counter to begin execution from the boot vector. This reset also clears the interrupt sticky bits.</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
2	RSVDMicro	R/W	<p>RSVDMicro: RSVD</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>
1	TestWakeup	R/W	<p>TestWakeup: Reserved. Must always be set to 0.</p> <p><b>Value After Reset:</b> 0x0  <b>Exists:</b> Always</p>

Bits	Name	Memory Access	Description
0	StallToMicro	R/W	<p>StallToMicro: This field is programmed as follows:</p> <ul style="list-style-type: none"><li>■ Set this bit to stall the microcontroller by hardware.</li><li>■ Clear this bit to allow the microcontroller to continue executing from its current program counter location.</li></ul> <p>Typically, this bit is used at power up to hold the program counter at the boot vector while BIOS loads the microcontroller program code. While stalled, the microcontroller clocks are gated off for power reduction.</p> <p><b>Value After Reset:</b> 0x1</p> <p><b>Exists:</b> Always</p>

### 13.8.31 MicroResetPIEEn

- **Description:** MicroResetPIEEn: Enables use PIE-specific ARC controls
- **Size:** 1 bit
- **Offset:** 0xd0000+0x9a
- **Exists:** Always



**Table 13-674 Fields for Register: MicroResetPIEEn**

Bits	Name	Memory Access	Description
0	MicroResetPIEEn	R/W	<p>MicroResetPIEEn: Must be set to use csrPIEStallToMicro and csrPIEResetToMicro</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.8.32 ClearPIESTallToMicro

- **Description:** ClearPIESTallToMicro: Forces csrPIESTallToMicro to 0
- **Size:** 1 bit
- **Offset:** 0xd0000+0x9b
- **Exists:** Always



**Table 13-675 Fields for Register: ClearPIESTallToMicro**

Bits	Name	Memory Access	Description
0	ClearPIESTallToMicro	R/W	<p>ClearPIESTallToMicro: A rising edge of this CSR forces csrPIESTallToMicro to 0. This bit does not auto-clear. Set it back low after setting it high.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p>

### 13.8.33 PIEMicroStallDelay

- **Description:** PIEMicroStallDelay: Delay before PIEStallToMicro takes effect
- **Size:** 4 bits
- **Offset:** 0xd0000+0x9c
- **Exists:** Always



**Table 13-676 Fields for Register: PIEMicroStallDelay**

Bits	Name	Memory Access	Description
3:0	PIEMicroStallDelay	R/W	<p>PIEMicroStallDelay: Specifies the number of DFICLKs before changes in the value of csrPIEStallToMicro will take effect</p> <p><b>Value After Reset:</b> 0x4</p> <p><b>Exists:</b> Always</p>

### 13.8.34 SequencerOverride

- **Description:** SequencerOverride: Reserved for PHY training firmware use.
- **Size:** 15 bits
- **Offset:** 0xd0000+0xe7
- **Exists:** Always



Table 13-677 Fields for Register: SequencerOverride

Bits	Name	Memory Access	Description
14:0	SequencerOverride	R/W	SequencerOverride: Reserved for Synopsys internal use <b>Value After Reset:</b> 0x80 <b>Exists:</b> Always

### 13.8.35 DfiInitCompleteShadow

- **Description:** DfiInitCompleteShadow: dfi\_init\_complete - Controller Read-only Shadow
- **Size:** 1 bit
- **Offset:** 0xd0000+0xfa
- **Exists:** Always



**Table 13-678 Fields for Register: DfiInitCompleteShadow**

Bits	Name	Memory Access	Description
0	DfiInitCompleteShadow	R	<p>DfiInitCompleteShadow: This csr presents a read-only view (a shadow) of the Register DfiInitComplete which is used by the sequencer to control the state of dfi_init_complete. The value in this Register is not affected by the BlockSeq0BAck field of the SequencerOverride register. While the Register MicroContMuxSel is set, access to this Shadow register will not steal config bus bandwidth from the micro controller. That is polling will not have a performance penalty.</p> <p><b>Value After Reset:</b> 0x0</p> <p><b>Exists:</b> Always</p> <p><b>Reset Mask:</b> 0x0</p> <p><b>Volatile:</b> true</p>

### 13.8.36 APBONLYReservedX (for X == 0)

- **Description:** APBONLYReservedX: Reserved for future use
- **Size:** 16 bits
- **Offset:** 0xd0000+0xfd
- **Exists:** Always



**Table 13-679 Fields for Register: APBONLYReservedX (for X == 0)**

Bits	Name	Memory Access	Description
15:0	APBONLYReserved0	R/W	APBONLYReserved0: Reserved for future use <b>Value After Reset:</b> 0x0 <b>Exists:</b> Always