DREX-1 v2.0.3

High-Performance SDRAM Controller

Revision 1.0

Aug. 2012

User's Guide

SAMSUNG ELECTRONICS

SAMSUNG

Important Notice

The information in this publication has been carefully checked and is believed to be entirely accurate at the time of publication. Samsung assumes no responsibility, however, for possible errors or omissions, or for any consequences resulting from the use of the information contained herein.

Samsung reserves the right to make changes in its products or product specifications with the intent to improve function or design at any time and without notice and is not required to update this documentation to reflect such changes.

This publication does not convey to a purchaser of semiconductor devices described herein any license under the patent rights of Samsung or others.

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

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

Samsung products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, for other applications intended to support or sustain life, or for any other application in which the failure of the Samsung product could create a situation where personal injury or death may occur.

Should the Buyer purchase or use a Samsung product for any such unintended or unauthorized application, the Buyer shall indemnify and hold Samsung and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, expenses, and reasonable attorney fees arising out of, either directly or indirectly, any claim of personal injury or death that may be associated with such unintended or unauthorized use, even if such claim alleges that Samsung was negligent regarding the design or manufacture of said product.

DREX-1 v2.0.3, High-Performance SDRAM Controller User's Guide, Revision 1.0

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

Contact Us: *taehun.kim@samsung.com*
TEL: (82)-(31)-209-4640
FAX: (82)-(31)-209-4494

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

Printed in the Republic of Korea

Revision History

Revision No.	Date	Description	Refer to	Author(s)
0.1	20111129	Wide IO memory		Kim Taehun
1.0	20120829	Beta release	All	Kim Taehun

Table of Contents

List of Figures

List of Tables

List of Conventions

Register RW Access Type Conventions

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

Register Value Conventions

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

Reset Value Conventions

Expression	Description
0	
1	
x	

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

List of Acronyms

Acronyms	Descriptions

1 DREX-1 V2.0.3

1.1 Overview

1.1.1 Introduction

The DREX-1 v2.0.31 is an Advanced Microcontroller Bus Architecture (AMBA) AXI-compliant slave to interface external JEDEC standard SDRAM devices. To support high-speed memory devices, the DREX-1 uses the DFI (DDR PHY Interface) as an interface protocol with SDRAM PHY. It features an advanced scheduler in an effort to transfer data to or from memory devices efficiently. Furthermore, its internal pipeline stages are optimized for higher performance and lower latency.

1.1.2 Features

- Compatible with JEDEC standard LPDDR3/DDR3 SDRAMs
- Supports 1:1 synchronous operation between AXI bus ACLK and scheduler CCLK
- Supports DFI 1:2 synchronous operation between ACLK/CCLK and memory clock domain
- Integrated TrustZone address space control unit
- Parameterized number of slave ports (1, 2, 3 or 4) compatible with AMBA3 AXI protocol for accesses to SDRAM devices
- One slave port compatible with AMBA3 APB protocol for programmable special function registers
- Another slave port compatible with AMBA3 APB protocol for programmable special function registers of the TrustZone address space control
- Uses the DFI SDRAM PHY interface to support high-speed memory devices
- Supports up to two memory ranks (chip selects) and 4/8 banks per memory chips
- Supports 1Gb, 2Gb, 4Gb and 8Gbit density per a chip select
- Supports QoS scheme to ensure low latency for real-time applications
- Out-of order scheduling policy for higher performance
- Detects AXI RAR/WAW hazards automatically
- Supports early write response
- Supports rank/bank interleaving
- Supports AMBA AXI low power interface for systemic power control
- Adapts to various low power schemes to reduce the dynamic and static current of memory
- Supports outstanding exclusive accesses
- Supports bank selective precharge policy

1 Hereafter, DREX-1 refers to DREX-1 v2.0.3 unless otherwise mentioned.

- Accommodates the embedded performance monitor
- ca_swap signal for reversing ca[9:0] to ca[0:9] for LPDDR2/LPDDR3
- Support Trust Zone Address Space Control

1.1.3 List of Supported SDRAM Devices

DREX-1 v2.0.3 supports the following SDRAM devices:

- LPDDR3 up to 800MHz
- DDR3

LPDDR2-S4 and Wide IO Memory is not supported in this version even though LPDDR2-S4 and Wide IO Memory is found below section.

1.1.4 Design Limitations

- Does not support locked access on all AMBA3 AXI/APB slave ports.
- Supports only DRAM data channel width of 32 bits.

1.2 Architecture Overview

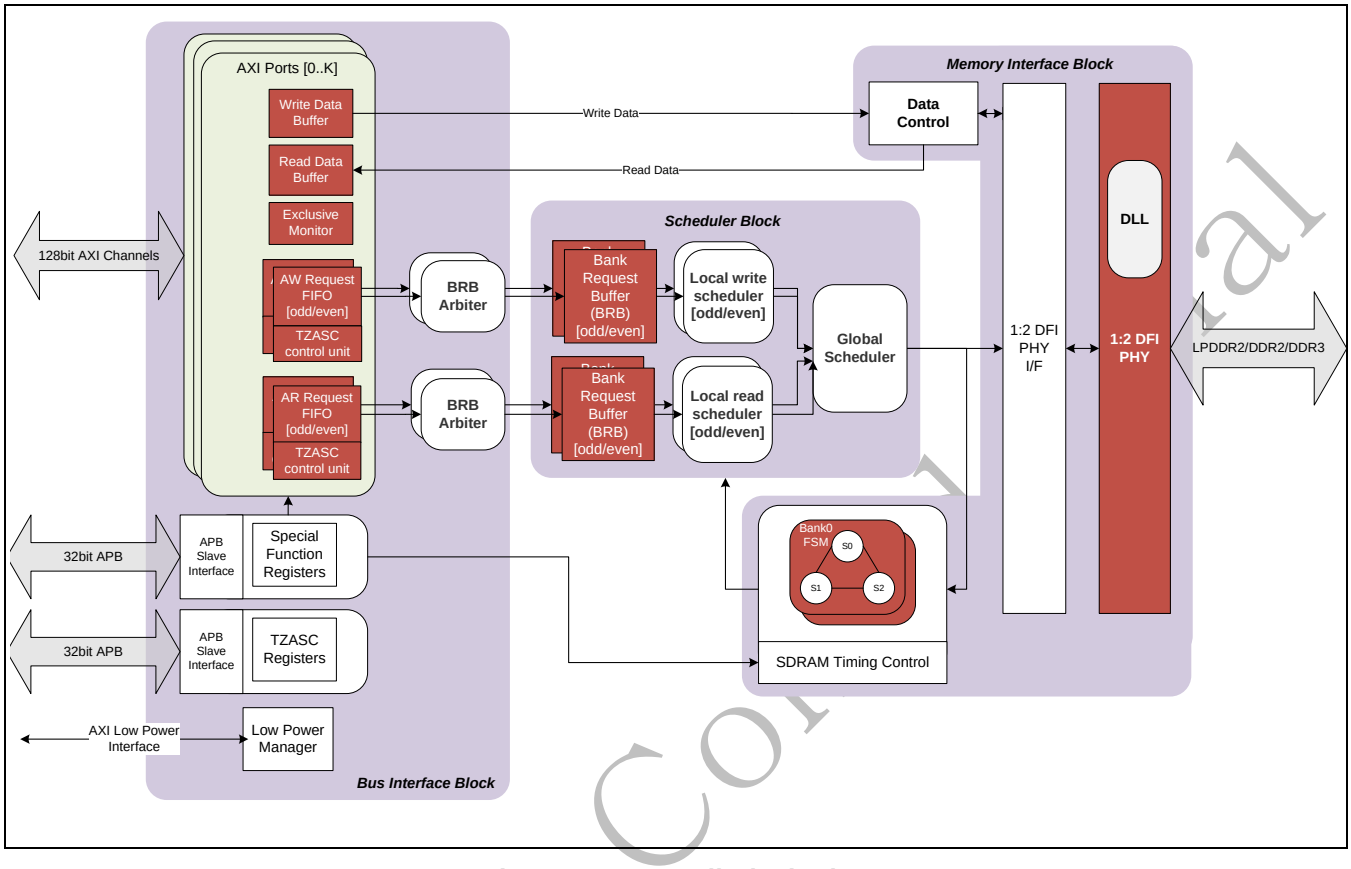

Figure 1-1 Overall Block Diagram

Figure 1-1 shows the overall block diagram of DREX-1. The block diagram shows the Bus Interface block, Scheduler Block, and Memory Interface Block, which connects and interfaces with DFI SDRAM PHYs.

The Bus Interface Block receives AXI transactions for memory access through an AXI slave port and inserts them into the request FIFO[2]. Meanwhile, it translates the address specified in an AXI transaction into memory address in the form of {rank, bank, row, column} format, and checks if the request is authorized to do the access according to the TZASC registers. Additionally, it stores incoming write data into the Write Data Buffer, whereas it sends read data retained within the Read Data Buffer to a master IP through the AXI slave port. The Read Data Buffer queues data when AXI masters are not ready to accept read data.

The APB slave interface for special function registers/direct commands and an AXI low power channel interface.

The Scheduler Block is where all the out-of-order operations happen. DREX-1 has a distributed scheduler which has 4 local schedulers (2 reads, 2 writes), each with their own request queues (Bank Request Buffers, or BRB), and a global scheduler which coordinates each of the local schedulers' output to generate requests to the PHY. The Scheduler Block users the SDRAM's state information from the Memory Interface Block to determine the optimal schedule for maximum performance. Meanwhile, it translates each memory request in the request buffers into appropriate memory command (e.g. PRE, ACT, RD, RDA, WR, WRA) as required by the corresponding bank

2 Transactions are distributed to each BRB and FIFO based on the bank number (odd/even) of the access. For example, a read request with a bank number of 3 will be queued on the read 'odd' BRB.

state.

The Memory Interface Block translates and maintains the state of the DRAMs using multiple finite state machines (FSM). Additionally, it translates memory commands coming from the Scheduler Block into corresponding SDRAM pin level signals and sends them to the PHY via the PHY I/F. It also updates each memory bank state according to the memory command and sends the bank state back to the scheduler.

1.3 Initialization

SDRAM devices must be powered up and initialized in a predefined manner. Operational procedures other than those specified may result in undefined operation. An initialization procedure consists of three procedures such as PHY DLL initialization, setting controller register and memory initialization. For memory initialization, please refer to JEDEC specifications and datasheets of memory devices. According to the memory types, initialization sequences are as follows.

1.3.1 LPDDR2/3 with PHY V5

The following sequence should be used to initialize LPDDR2 devices. Unless specified otherwise, these steps are mandatory.

1. To provide stable power for memory device, DREX must assert and hold CKE to a logic low level. Then apply stable clock.
2. Set the right value to PHY control register0 for LPDDR2/3 operation mode. If read leveling is needed, check LPDDR2/3 IO calibration MRR data and match it to PHY control register1's ctrl_rlvl_rdata_adj field. (Refer to PHY manual)
3. Set the PHY for dqs pulldown mode. (Refer to PHY manual, PHY control register 14)
4. Assert the **ConControl.dfi_init_start** field to high but leave as default value for other fields.(aref_en and io_pd_con should be off.) Clock gating in **CGControl** should be disabled in initialization and training sequence.
5. Wait for the **PhyStatus0.dfi_init_complete** field to change to '1'.
6. Deassert the **ConControl.dfi_init_start** field to low.
7. Set the **PhyControl0.fp_resync** bit-field to '1' to update DLL information.
8. Set the **PhyControl0.fp_resync** bit-field to '0'.
9. Set the **MemBaseConfig0** MemBaseConfig0 register. If there are two external memory chips, set the **MemBaseConfig1** register.
10. Set the **MemConfig0** register. If there are two external memory chips, also set the **MemConfig1** register.
11. Set the **PrechConfig0/1** and **PwrdnConfig** registers.
12. Set the **TimingAref, TimingRow, TimingData** and **TimingPower** registers according to memory AC parameters.
13. If QoS scheme is required, set the **QosControl0~15** and **QosConfig0~15** registers.
14. Set the PHY for LPDDR2/3 initialization. LPDDR2/3 initialization clock period is 18ns ~ 100ns. If LPDDR2/3 initialization clock period is 20ns, then follow below steps 15 ~ 21(refer to PHY manual).
15. Set the PHY ctrl_offsetr0~3 and ctrl_offsetw0~3 value to 0x7F
16. Set the PHY ctrl_offsetd value to 0x7F.
17. Set the PHY ctrl_force value to 0x7F.
18. Set the PHY ctrl_dll_on to low.
19. Wait for 10 PCLK cycles.
20. Set the **PhyControl0.fp_resync** bit-field to '1' to update DLL information.
21. Set the **PhyControl0.fp_resync** bit-field to '0'.
22. Confirm that CKE has been as a logic low level at least 100ns after power on
23. Issue a NOP command using the **DirectCmd** register to assert and to hold CKE to a logic high level.
24. Wait for minimum 200us.
25. Issue a MRS command using the **DirectCmd** register to reset memory devices and program the operating parameters.
26. Wait for minimum 10us.
27. Issue proper lpddr2 initialization commands according to specification such as IO calibration (MR #10), device feature1,2 (MR #1, #2). Refer to LPDDR2/3 specification for details.
28. If there are two external memory chips, perform steps 23 ~ 27 for chip1 memory device.

29. Set the PHY ctrl_offsetr0~3 and ctrl_offsetw0~3 value to 0x0
30. Set the PHY ctrl_offsetd value to 0x0
31. Set the PHY ctrl_dll_on enable
32. Wait for 10 PCLK cycles.
33. Set the PHY ctrl_start value to '0'.
34. Set the PHY ctrl_start value to '1'.
35. Wait for 10 PCLK cycles.
36. Wait for the **PhyStatus0.dfi_init_complete** field to change to ‘1’.
37. Set the **PhyControl0.fp_resync** bit-field to ‘1’ to update DLL information.
38. Set the **PhyControl0.fp_resync** bit-field to ‘0’.
39. If any leveling/training is needed, disable ctrl_dll_on and set ctrl_force value. (Refer to PHY manual)
40. If write leveling for LPDDR3 is not needed, skip this procedure. If write leveling is needed, set LPDDR3 into write leveling mode using MRS command, set ODT pin high and tWLO using WRLVL_CONFIG0 register(offset=0x120) and set the related PHY SFR fields through PHY APB I/F(Refer to PHY manual). To generate 1 cycle pulse of dfi_wrdata_en_p0, write 0x1 to WRLVL_CONFIG1 register(Offset addr=0x124). To read the value of memory data, use CTRL_IO_RDATA(offset = 0x150). If write leveling is finished, then set ODT pin low and disable write leveling mode of LPDDR3
41. If CA calibration for LPDDR3 is not needed, skip this procedure. If CA calibration is needed, set the related PHY SFR fields through PHY APB I/F(Refer to PHY manual). Set LPDDR3 into CA calibration mode through MR41. For CKE assertion, deassetion, CA value and tADDR setting, use CACAL_CONFIG0(offset = 0x160). For Generation 1 cycle pulse of dfi_csn_p0, use CACAL_CONFIG1(offset = 0x164). To read the value of memory data, use CTRL_IO_RDATA (offset = 0x150). Note that CKE pin should be asserted when MR41/48/42 command is issued and CKE pin should be deasserted during CA calibration. Also note that because CA SDLL code updating needs some cycles to complete, 10 PCLK cycles are needed before issuing next command.
42. If read leveling is not needed, skip 43 ~ 47. If read leveling is needed, set the related PHY SFR fields through PHY APB I/F. The related register is PHY control register #1 and #2(mpr value, ctrl_rdlvl_gate_en and ctrl_rdlvl_en register, refer to PHY manual).
43. Set the **RdlvlConfig.ctrl_rdlvl_data_en** bit-field to 1'b1. Gate training is not supported.
44. Wait for the **PhyStatus0.read_level_complete** field to change to ‘1’.
45. Disable the RdlvlConfig.ctrl_rdlvl_gate_en and ctrl_rdlvl_data_en.
46. Set the **PhyControl0.fp_resync** bit-field to ‘1’ to update DLL information.
47. Set the **PhyControl0.fp_resync** bit-field to ‘0’.
48. If write training is not needed, skip 49 ~ 54. If write training is nedded, set the related PHY SFR fields through PHY APB I/F. The related register is PHY control register #2 and #26, refer to PHY manual)
49. Set write latency of PHY control register #26.
50. Enable **WrtraConfig.write_training_en** to issue ACT command. Refer to this register definition for row and bank address.
51. Wait for 10 PCLK cycles.
52. Enable write de-skewing of PHY control register #2.
53. Wait for the **PhyStatus0.read_level_complete** field to change to ‘1’.
54. Disable **WrtraConfig.write_training_en**.
55. Disable write de-skewing of PHY control register #2.
56. After all leveling/training are completed, enable ctrl_dll_on. (Refer to PHY manual)

57. Set the **PhyControl0.fp_resync** bit-field to '1' to update DLL information.
58. Set the **PhyControl0.fp_resync** bit-field to '0'.
59. Disable PHY gating control through PHY APB I/F if necessary(ctrl_atgate, see PHY manual).
60. Issue PALL to all chips using direct command. This is an important step if write training has been done.
61. Set the **MemControl and PhyControl0** register.
62. Set the **ConControl** register. aref_en should be turn on.
63. Set the **CGControl** register for clock gating enable.

1.3.2 LPDDR2/3 with PHY V6

The following sequence should be used to initialize LPDDR2 devices. Unless specified otherwise, these steps are mandatory.

1. To provide stable power for memory device, DREX must assert and hold CKE to a logic low level. Then apply stable clock.
2. Set the PHY for DDR3 operation mode, RL/WL/BL register and proceed ZQ calibration. Refer to "INITIALIZATION" in PHY manual.
3. Assert the **ConControl.dfi_init_start** field to high but leave as default value for other fields.(aref_en and io_pd_con should be off.) Clock gating in **CGControl** should be disabled in initialization and training sequence.
4. Wait for the **PhyStatus0.dfi_init_complete** field to change to '1'.
5. Deassert the **ConControl.dfi_init_start** field to low.
6. Set the PHY for dqs pulldown mode. (Refer to PHY manual)
7. Set the **PhyControl0.fp_resync** bit-field to '1' to update DLL information.
8. Set the **PhyControl0.fp_resync** bit-field to '0'.
9. Set the **MemBaseConfig0** MemBaseConfig0 register. If there are two external memory chips, set the **MemBaseConfig1** register.
10. Set the **MemConfig0** register. If there are two external memory chips, also set the **MemConfig1** register.
11. Set the **PrechConfig0/1** and **PwrdnConfig** registers.
12. Set the **TimingAref, TimingRow, TimingData** and **TimingPower** registers according to memory AC parameters.
13. If QoS scheme is required, set the **QosControl0~15** and **QosConfig0~15** registers.
14. Set the PHY for LPDDR2/3 initialization. LPDDR2/3 initialization clock period is 18ns ~ 100ns. If LPDDR2/3 initialization clock period is 20ns, then follow below steps 15 ~ 21(refer to PHY manual).
15. Set the PHY ctrl_offsetr0~3 and ctrl_offsetw0~3 value to 0x7F
16. Set the PHY ctrl_offsetd value to 0x7F.
17. Set the PHY ctrl_force value to 0x7F.
18. Set the PHY ctrl_dll_on to low.
19. Wait for 10 PCLK cycles.
20. Set the **PhyControl0.fp_resync** bit-field to '1' to update DLL information.
21. Set the **PhyControl0.fp_resync** bit-field to '0'.
22. Confirm that CKE has been as a logic low level at least 100ns after power on
23. Issue a NOP command using the **DirectCmd** register to assert and to hold CKE to a logic high level.
24. Wait for minimum 200us.
25. Issue a MRS command using the **DirectCmd** register to reset memory devices and program the operating parameters.
26. Wait for minimum 10us.
27. Issue proper lpddr2 initialization commands according to specification such as IO calibration (MR #10), device feature1,2 (MR #1, #2). Refer to LPDDR2/3 specification for details.
28. If there are two external memory chips, perform steps 23 ~ 27 for chip1 memory device.
29. Set the PHY ctrl_offsetr0~3 and ctrl_offsetw0~3 value to 0x0

30. Set the PHY ctrl_offsetd value to 0x0
31. Set the PHY ctrl_dll_on enable
32. Wait for 10 PCLK cycles.
33. Set the PHY ctrl_start value to '0'.
34. Set the PHY ctrl_start value to '1'.
35. Wait for 10 PCLK cycles.
36. Wait for the **PhyStatus0.dfi_init_complete** field to change to ‘1’.
37. Set the **PhyControl0.fp_resync** bit-field to ‘1’ to update DLL information.
38. Set the **PhyControl0.fp_resync** bit-field to ‘0’.
39. If any leveling/training is needed, enable ctrl_atgate, p0_cmd_en, InitDeskewEn and byte_rdlvl_en. Disable ctrl_dll_on and set ctrl_force value. (Refer to PHY manual)
40. If write leveling for LPDDR3 is not needed, skip this procedure. If write leveling is needed, set LPDDR3 into write leveling mode using MRS command, set ODT pin high and tWLO using WRLVL_CONFIG0 register(offset=0x120) and set the related PHY SFR fields through PHY APB I/F(Refer to PHY manual). To generate 1 cycle pulse of dfi_wrdata_en_p0, write 0x1 to WRLVL_CONFIG1 register(Offset addr=0x124). To read the value of memory data, use CTRL_IO_RDATA(offset = 0x150). If write leveling is finished, then set ODT pin low and disable write leveling mode of LPDDR3
41. If CA calibration for LPDDR3 is not needed, skip this procedure. If CA calibration is needed, set the related PHY SFR fields through PHY APB I/F(Refer to PHY manual). Set LPDDR3 into CA calibration mode through MR41. For CKE assertion, deassetion, CA value and tADDR setting, use CACAL_CONFIG0(offset = 0x160). For Generation 1 cycle pulse of dfi_csn_p0, use CACAL_CONFIG1(offset = 0x164). To read the value of memory data, use CTRL_IO_RDATA_CH0/CH1 (offset = 0x150, 0x154). Note that CKE pin should be asserted when MR41/48/42 command is issued and CKE pin should be deasserted during CA calibration. Also note that because CA SDLL code updating needs some cycles to complete, 10 PCLK cycles are needed before issuing next command.
42. If read leveling is not needed skip this procedure. If read leveling is needed, set proper value to PHY control register. Do the read leveling..(Refer to PHY manual)
43. If write training is not needed, skip this procedure. If write training is nedded, set the related PHY SFR fields through PHY APB I/F.). To issue ACT command, enable and disable **WrtraConfig.write_training_en .** Refer to this register definition for row and bank address. Do write training. (Refer to PHY manual)
44. After all leveling/training are completed, enable ctrl_dll_on. (Refer to PHY manual)
45. Set the **PhyControl0.fp_resync** bit-field to ‘1’ to update DLL information.
46. Set the **PhyControl0.fp_resync** bit-field to ‘0’.
47. Disable PHY gating control through PHY APB I/F if necessary(ctrl_atgate, see PHY manual).
48. Issue PALL to all chips using direct command. This is an important step if write training has been done.
49. Set the **MemControl and PhyControl0** register.
50. Set the **ConControl** register. aref_en should be turn on.
51. Set the **CGControl** register for clock gating enable.

1.3.3 DDR3 with PHY V5

The following sequence should be used to initialize DDR3 devices. Unless specified otherwise, these steps are mandatory.

1. Apply power. RESET# pin of memory needs to be maintained for minimum 200us with stable power. CKE is pulled “Low” anytime before RESET# being de-asserted (min. time 10ns)
2. Set the PHY for DDR3 operation mode. . If read leveling is needed, check DDR3 MPR data and match it to PHY control register1's ctrl_rlvl_rdata_adj field. (Refer to PHY manual)
3. Set the PHY for dqs pulldown mode. (Refer to PHY manual, PHY control register 14)
4. If on die termination is required, enable **PhyControl0.mem_term_en**, **PhyControl0.phy_term_en**.
5. Assert the **ConControl.dfi_init_start** field to high but leave as default value for other fields.(aref_en and io_pd_con should be off.) Clock gating in **CGControl** should be disabled in initialization and training sequence.
6. Wait for the **PhyStatus0.dfi_init_complete** field to change to ‘1’.
7. Deassert the **ConControl.dfi_init_start** field to low.
8. Set the **PhyControl0.fp_resync** bit-field to ‘1’ to update DLL information.
9. Set the **PhyControl0.fp_resync** bit-field to ‘0’.
10. Set the **MemBaseConfig0** MemBaseConfig0 register. If there are two external memory chips, set the **MemBaseConfig1** register.
11. Set the **MemConfig0** register. If there are two external memory chips, also set the **MemConfig1** register.
12. Set the **PrechConfig0/1** and **PwrdnConfig** registers.
13. Set the **TimingAref**, **TimingRow**, **TimingData** and **TimingPower** registers according to memory AC parameters.
14. If QoS scheme is required, set the **QosControl0~15** and **QosConfig0~15** registers.
15. Confirm that after RESET# is de-asserted, 500 us have passed before CKE becomes active.
16. Confirm that clocks(CK, CK#) need to be started and stabilized for at least 10 ns or 5 tCK (which is larger) before CKE goes active.
17. Issue a NOP command using the **DirectCmd** register to assert and to hold CKE to a logic high level.
18. Wait for tXPR(max(5nCK,tRFC(min)+10ns)) or set tXP to tXPR value before step 17. If the system set tXP to tXPR, then the system must set tXP to proper value before normal memory operation.
19. Issue an EMRS2 command using the **DirectCmd** register to program the operating parameters. Dynamic ODT should be disabled. A10 and A9 should be low.
20. Issue an EMRS3 command using the **DirectCmd** register to program the operating parameters.
21. Issue an EMRS command using the **DirectCmd** register to enable the memory DLL.
22. Issue a MRS command using the **DirectCmd** register to reset the memory DLL.
23. Issues a MRS command using the **DirectCmd** register to program the operating parameters without resetting the memory DLL.
24. Issues a ZQINIT commands using the **DirectCmd** register.
25. If there are two external memory chips, perform steps 17 ~ 24 procedures for chip1 memory device.
26. If any leveling/training is needed, disable ctrl_dll_on and set ctrl_force value. (Refer to PHY manual)
27. If write leveling is not needed, skip this procedure. If write leveling is needed, set DDR3 into write leveling mode using MRS command, set ODT pin high and tWLO using WRLVL_CONFIG0 register(offset=0x120) and set the related PHY SFR fields through PHY APB I/F(Refer to PHY manual). To generate 1 cycle pulse of

dfi_wrdata_en_p0, write 0x1 to WRLVL_CONFIG1 register(Offset addr=0x124). To read the value of memory data, use CTRL_IO_RDATA(offset = 0x150). If write leveling is finished, then set ODT pin low and disable write leveling mode of DDR3

28. If gate leveling is not needed, skip 29 ~ 32. Gate leveling is only supported DDR3 over 667Mhz. If gate leveling is needed, set the related PHY SFR fields through PHY APB I/F. The related register is PHY control register #0, #1 and #2(Refer to PHY manual)
29. Set the **RdlvlConfig.ctrl_rdlvl_gate_en** bit-field to 1'b1.
30. Wait for the **PhyStatus0.read_level_complete** field to change to '1'.
31. Disable DQS pulldown mode.(Refer to PHY manual)
32. Disable the RdlvlConfig.ctrl_rdlvl_gate_en and ctrl_rdlvl_data_en.
33. If read leveling is not needed, skip **오류! 참조 원본을 찾을 수 없습니다.** ~ 38. If read leveling is needed, set the related PHY SFR fields through PHY APB I/F. The related register is PHY control register #1 and #2(mpr value, ctrl_rdlvl_gate_en and ctrl_rdlvl_en register, refer to PHY manual).
34. Set the **ctrl_rdlvl_data_en** bit-field to 1'b1. MPR command is issued automatically.
35. Wait for the **PhyStatus0.read_level_complete** field to change to '1'.
36. Disable the RdlvlConfig.ctrl_rdlvl_gate_en and ctrl_rdlvl_data_en.
37. Set the **PhyControl0.fp_resync** bit-field to '1' to update DLL information.
38. Set the **PhyControl0.fp_resync** bit-field to '0'.
39. If write training is not needed, skip 40 ~ 45. If write training is nedded, set the related PHY SFR fields through PHY APB I/F. The related register is PHY control register #2 and #26, refer to PHY manual)
40. Set write latency of PHY control register #26.
41. Enable **WrtraConfig.write_training_en** to issue ACT command. Refer to this register definition for row and bank address.
42. Wait for 10 PCLK cycles.
43. Enable write de-skewing of PHY control register #2.
44. Wait for the **PhyStatus0.read_level_complete** field to change to '1'.
45. Disable **WrtraConfig.write_training_en**.
46. Disable write de-skewing of PHY control register #2.
47. After all leveling/training are completed, enable ctrl_dll_on. (Refer to PHY manual)
48. Set the **PhyControl0.fp_resync** bit-field to '1' to update DLL information.
49. Set the **PhyControl0.fp_resync** bit-field to '0'.
50. Disable PHY gating control through PHY APB I/F if necessary(ctrl_atgate, refer to PHY manual).
51. Issue PALL to all chips using direct command. This is an important step if write training has been done.
52. Set the **MemControl and PhyControl0** register.
53. Set the **ConControl** register. aref_en should be turn on.
54. Set the **CGControl** register for clock gating enable.

1.3.4 DDR3 with PHY V6

1. Apply power. RESET# pin of memory needs to be maintained for minimum 200us with stable power. CKE is pulled “Low” anytime before RESET# being de-asserted (min. time 10ns)
2. Set the PHY for DDR3 operation mode, RL/WL/BL register and proceed ZQ calibration. Refer to "INITIALIZATION" in PHY manual.
3. Assert the **ConControl.dfi_init_start** field to high but leave as default value for other fields.(aref_en and io_pd_con should be off.) Clock gating in **CGControl** should be disabled in initialization and training sequence.
4. Wait for the **PhyStatus0.dfi_init_complete** field to change to ‘1’.
5. Deassert the **ConControl.dfi_init_start** field to low.
6. Set the PHY for dqs pulldown mode. (Refer to PHY manual)
7. Set the **PhyControl0.fp_resync** bit-field to ‘1’ to update DLL information.
8. Set the **PhyControl0.fp_resync** bit-field to ‘0’.
9. Set the **MemBaseConfig0** register and MemBaseConfig1 register if needed.
10. Set the **MemConfig0** register and MemConfig1 if needed..
11. Set the **PrechConfig** and **PwrdnConfig** registers.
12. Set the **TimingAref**, **TimingRow**, **TimingData** and **TimingPower** registers according to memory AC parameters.
13. If QoS scheme is required, set the **QosControl0~15** and **QosConfig0~15** registers.
14. Confirm that after RESET# is de-asserted, 500 us have passed before CKE becomes active.
15. Confirm that clocks(CK, CK#) need to be started and stabilized for at least 10 ns or 5 tCK (which is larger) before CKE goes active.
16. Issue a NOP command using the **DirectCmd** register to assert and to hold CKE to a logic high level.
17. Wait for tXPR(max(5nCK,tRFC(min)+10ns)) or set tXP to tXPR value before step 17. If the system set tXP to tXPR, then the system must set tXP to proper value before normal memory operation.
18. Issue an EMRS2 command using the **DirectCmd** register to program the operating parameters. Dynamic ODT should be disabled. A10 and A9 should be low.
19. Issue an EMRS3 command using the **DirectCmd** register to program the operating parameters.
20. Issue an EMRS command using the **DirectCmd** register to enable the memory DLL.
21. Issue a MRS command using the **DirectCmd** register to reset the memory DLL.
22. Issues a MRS command using the **DirectCmd** register to program the operating parameters without resetting the memory DLL.
23. Issues a ZQINIT commands using the **DirectCmd** register.
24. If there are more external memory chips, perform steps 17 ~ 24 procedures for other memory device.
25. If any leveling/training is needed, enable ctrl_atgate, p0_cmd_en, InitDeskewEn and byte_rdlvl_en. Disable ctrl_dll_on and set ctrl_force value. (Refer to PHY manual)
26. If write leveling is not needed, skip this procedure. If write leveling is needed, set DDR3 into write leveling mode using MRS direct command, set ODT pin high and tWLO using WRLVL_CONFIG0 register(offset=0x120) and set the related PHY SFR fields through PHY APB I/F(Refer to PHY manual). To generate 1 cycle pulse of dfi_wrdata_en_p0, write 0x1 to WRLVL_CONFIG1 register(Offset addr=0x124). To read the value of memory data, use CTRL_IO_RDATA(offset = 0x150). If write leveling is finished, disable write leveling mode in PHY register and set ODT pin low and disable write leveling mode of DDR3.

27. If gate leveling is not needed, skip 27 ~ 28. If gate leveling is needed, set DDR3 into MPR mode using MRS direct command and set the related PHY SFR fields through PHY APB I/F. Do the gate leveling. (Refer to PHY manual)
28. If gate leveling is finished, set DDR3 into normal operation mode using MRS command and disable DQS pull-down mode.(Refer to PHY manual)
29. If read leveling is not needed skip 29 ~ 30. If read leveling is needed, set DDR3 into MPR mode using MRS direct command and set proper value to PHY control register. Do the read leveling..(Refer to PHY manual)
30. If read leveling is finished, set DDR3 into normal operation mode using MRS direct command.
31. If write training is not needed, skip 31. If write training is nedded, set the related PHY SFR fields through PHY APB I/F..(Refer to PHY manual). To issue ACT command, enable and disable **WrtraConfig.write_training_en .** Refer to this register definition for row and bank address. Do write training. (Refer to PHY manual)
32. After all leveling/training are completed, enable ctrl_dll_on. (Refer to PHY manual)
33. Set the **PhyControl0.fp_resync** bit-field to '1' to update DLL information.
34. Set the **PhyControl0.fp_resync** bit-field to '0'.
35. Disable PHY gating control through PHY APB I/F if necessary(ctrl_atgate, refer to PHY manual).
36. Issue PALL to all chips using direct command. This is an important step if write training has been done.
37. Set the **MemControl and PhyControl0** register.
38. Set the **ConControl** register. aref_en should be turn on.
39. Set the **CGControl** register for clock gating enable.

1.3.5 Wide IO SDR

The following sequence should be used to initialize Wide IO SDR devices. Unless specified otherwise, these steps are mandatory.

1. To provide stable power for memory device, DREX must assert and hold CKE to a logic high level. Then apply stable clock .
2. Set **PHYCONTROL0.dqs_delay** to 1.
3. Set the PHY register(ctrl_start_point,ctrl_inc) bit-fields to correct value according to clock frequency. Set the PHY register(ctrl_dll_on) bit-field to '1' to activate the PHY DLL.
4. DQS Cleaning : set the PHY register(ctrl_shiftc,ctrl_offsetc) bit-fields to the proper value according to clock frequency, board delay and memory tDQSCK parameter.
5. Set the PHY register(ctrl_start) bit-field to '1'.
6. Set the **ConControl**. At this moment, aref_en and io_pd_con fields should be off. Clock gating in **CGControl** should be disabled in initialization and training sequence.
7. Set the **MemControl**. At this moment, all power down modes including sl_dll_dyn_con should be off.
8. Set the **MemBaseConfig0** MemBaseConfig0 register. If there are two external memory chips, set the **MemBaseConfig1** register.
9. Set the **MemConfig0** register. If there are two external memory chips, also set the **MemConfig1** register.
10. Set the **PrechConfig0/1** and **PwrdnConfig** registers.
11. Set the **TimingAref**, **TimingRow**, **TimingData** and **TimingPower** registers according to memory AC parameters.
12. If QoS scheme is required, set the **QosControl0~15** and **QosConfig0~15** registers.
13. Wait for the PHY register(ctrl_clock,ctrl_flock) bit-fields to change to '1'. Check whether PHY DLL is locked.
14. PHY DLL compensates the changes of delay amount caused by PVT variation during memory operation. Therefore, it should not be off for reliable operation. It can be off except runs at low frequency. If off mode is used, set the PHY register(ctrl_force) bit-field to the correct value according to the PHY register(ctrl_lock_value)[9:2] bit-field for fix delay amount. Clear the PHY register(ctrl_dll_on) bit-field to turn off PHY DLL.
15. Confirm that CKE has been as a logic low level at least 100ns after power on.
16. Confirm whether stable clock issues minimum 5 tCK before first CKE high.
17. Confirm minimum 200 us idle time after first CKE assertion.
18. Isue a NOP command using the **DirectCmd** register.
19. Issue a PALL command using the **DirectCmd** register.
20. Issue two Auto Refresh commands using the **DirectCmd** register.
21. Issue a MRS command using the **DirectCmd** register to program the operating parameters.
22. Issue an EMRS command using the **DirectCmd** register to program the operating parameters.
23. If there are two external memory chips, perform steps 15 ~ 22steps for chip1 memory device.
24. Set the **ConControl** to turn on an auto refresh counter
25. If power down mode is required, set the **MemControl** register.

1.4 Address Mapping

DREX-1 modifies the address of the AXI transaction coming from the AXI slave port into a memory address – chip select, bank address, row address, column address and memory data width. 'width' represents the data width of the DRAM used, which is fixed to 32 bits(4 bytes), and hence, on a byte-addressed address value, the corresponding width is fixed to 2.

The related SFR is bank_lsb, chip_inter_en, bit_sel_en, chip_map in MemConfig0/1 registers.

In case of chip interleaving, two chips configuration should the same.

To map chip select of memory device to a specific area of the address map, the **chip_base** and **chip_mask** bit-fields of the **MemConfig0** register needs to be set (Refer to Register Descriptions). If chip1 of the memory device exists, the **MemConfig1** register must also be set. Then, the AXI address requested by AXI Masters is divided into the AXI base address and AXI offset address. The AXI base address activates the appropriate memory chip select and the AXI offset address is mapped to a memory address according to the bank, row, column number, and data width set by the **MemConfig0/1** and **MemControl** register.

There are two ways to map the AXI offset address as shown below: 1) simple interleaved mapping 2) split column interleaved mapping 3) randomized interleaved mapping 4) chip interleaved mapping

1.4.1 Split column interleaved mapping

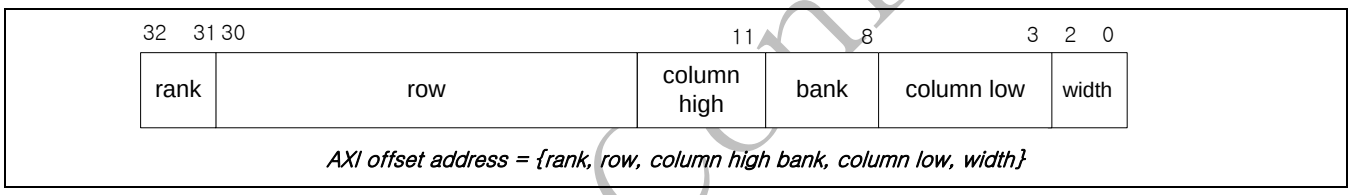

Figure 1-2 Split column Interleaved Address Mapping

chip_inter_en = 0x0, bit_sel_en = 0x0 and chip_map = 0x2 means split column interleaved mapping.

As shown in *Figure 1-2* the split column interleaved mapping method maps the AXI address in the order of row, column high, bank, column low and width.

The related SFR is "bank_lsb" which select column low size.(Refer to Register Descriptions). If bank_lsb is the same with the actual memory page size, then column would not be splited.

1.4.2 Randomized interleaved mapping

In addition to split column interleaved address mapping, DREX supports randomized interleaved mapping. As shown in Figure 1-3, bank address is randomized by XORing with additional bits chosen from the AXI address. It further improves performance by distributing memory accesses to numerous banks more aggressively.

chip_inter_en = 0x0 and bit_sel_en = 0x1 means randomized interleaved mapping.

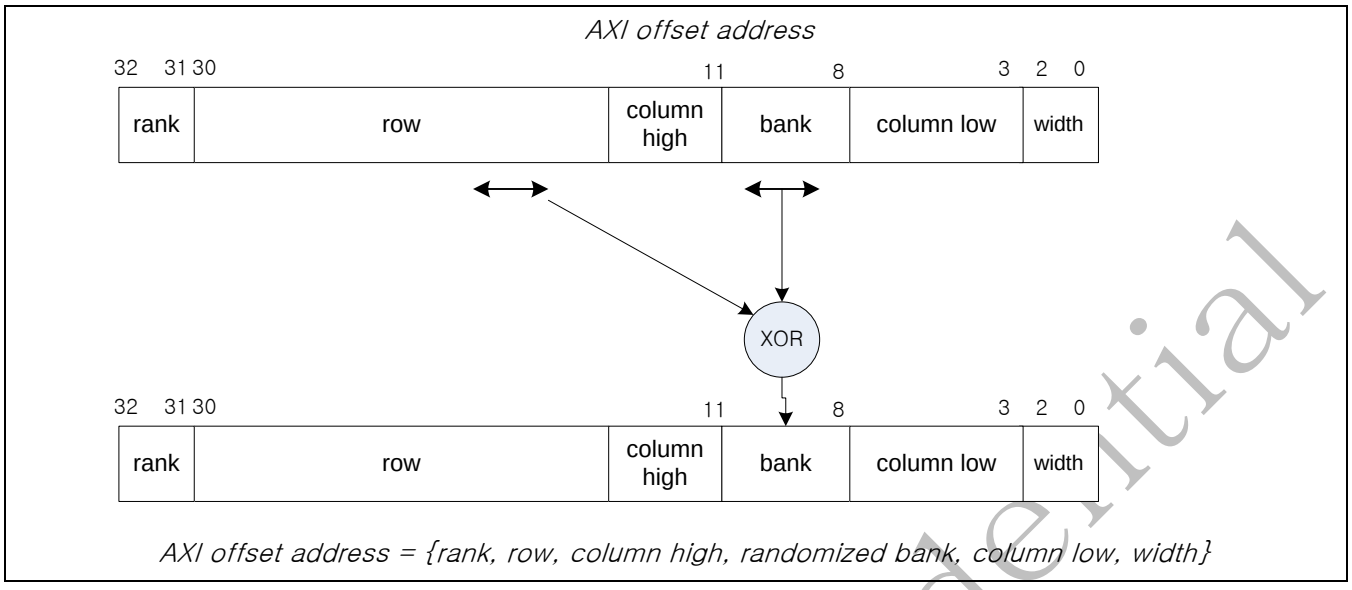

Figure 1-3 Randomized interleaved Address Mapping

1.4.3 Rank interleaved mapping

rank_inter_en = 0x1 means rank interleaved mapping. If rank_inter_en is enabled, then above figure are changed to below repectively.

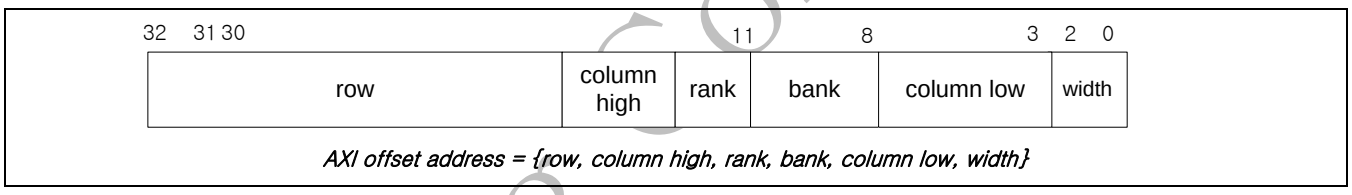

Figure 1-4 Rank Interleaved Address Mapping (chip_map = 0x2, bit_sel_en = 0x0)

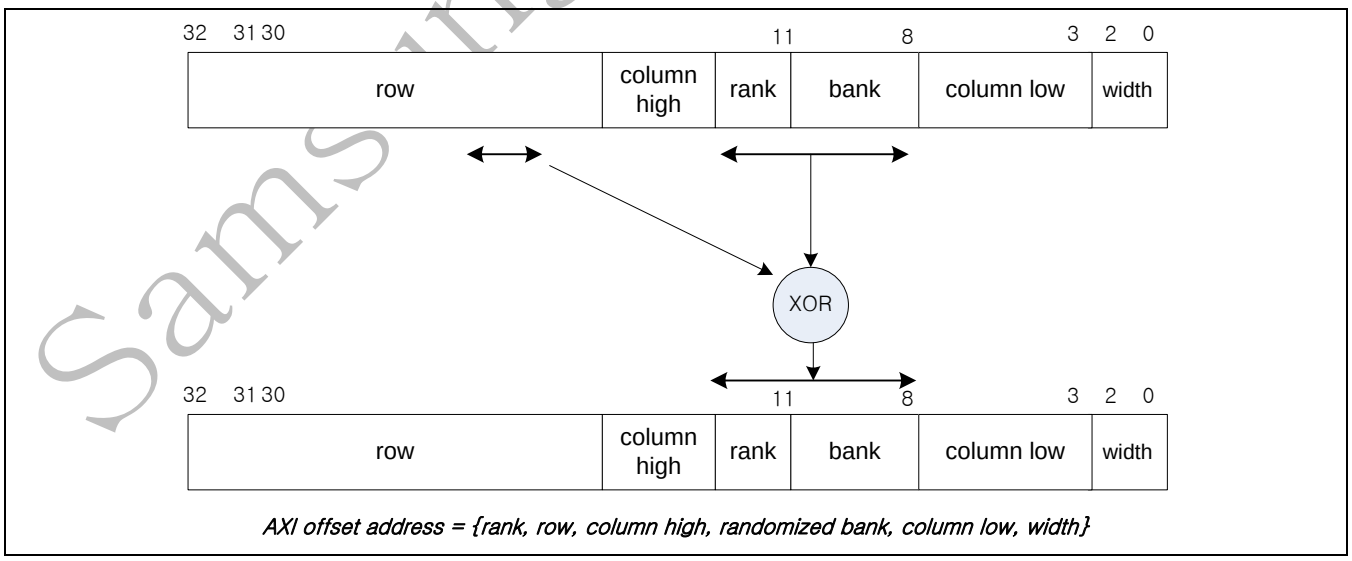

Figure 1-5 Rank Interleaved Address Mapping (chip_map = 0x2, bit_sel_en = 0x1)

1.5 Low Power Operation

The controller executes a low power memory operation in five ways as described below. Each feature is independent of each other and executed at the same time.

When memory is in the SREF state, then the controller issue SRX command automatically if AXI request is comming.

1.5.1 AXI low-power interface

The controller has an AXI low-power interface to communicate with low power management units such as the system controller, which makes the memory device go into self refresh mode.

1.5.2 Dynamic power down

An SDRAM device has an active/precharge power down mode. This mode is triggered by deasserting CKE to LOW. When any of the banks is open, it enters active power down mode. Otherwise, it enters precharge power down mode.

When the request buffers remain empty for certain number of cycles (PwrdnConfig.dpwrdn_cyc register), DREX-1 changes the memory device's state to active/precharge power down automatically. The memory device enters Active/precharge power down mode or Forced precharge power down mode according to the SFR setting. The description of the two power down modes are as follows:

1. Active/precharge power down mode: Enter power down w/o considering whether there is a row open or not.
2. Forced precharge power down mode: Enter power down after closing all banks.

When DREX-1 receives a new AXI transaction while memory device is in power down mode, it automatically wakes up the memory device from power down state and executes in a normal operation state.

1.5.3 Dynamic self refresh

Similarly to the dynamic power down feature, if the request buffers remain empty for certain number of cycles (PwrdnConfig.dsref_cyc register), DREX-1 changes the memory device's state to self-refresh mode. Since exiting power down mode requires many cycles, a longer idle cycle threshold is recommended for dynamic self-refresh entry than the threshold for dynamic power down.

1.5.4 Clock stop

To reduce the I/O power of the memory device and the controller, it is possible to stop the clock if the LPDDR / LPDDR2-S4 is in idle mode, or self refresh mode and DDR2/DDR3 is in self refresh mode. If this feature is enabled, the controller automatically executes the clock stop feature. In DDR3, clock stop feature must be turn-on and off considering tCKSRX/tCKSRE/tCKESR timing by software.

1.5.5 Direct command

Use the direct command feature to send a memory command directly to the memory device through the APB3 port. This way, it is possible to force the memory device to enter active/precharge power down, self-refresh mode.

1.6 Precharge Policy

There are two options for DREX-1 regarding precharge policy – port-selective precharge and timeout precharge per port.

1.6.1 Port Selective Precharge

Since applications have different page policy preferences, it is hard for the engineer to decide on whether to use open page policy, or close page (auto precharge) policy. Instead of applying the page policy to all of the ports, the port selective precharge policy allows the user to choose a precharge policy for each port (refer to **PrechConfig.port_policy**). This way, you assign certain applications to an open page policy, and other applications to a close page (auto precharge) policy.

- Open Page Policy: After a READ or WRITE, the accessed row is left open.

- Close Page (Auto Precharge) Policy: When DREX-1 issues the last READ or WRITE CAS command, it augments the command with an auto precharge flag.

1.6.2 Timeout precharge

If a certain port uses an open page policy, the row is left open after a data access. If this happens and the bank that is left open is not scheduled for a specific number of cycles (**PrechConfig.tp_cnt** bit-field) the controller automatically issues a precharge command to close the bank.

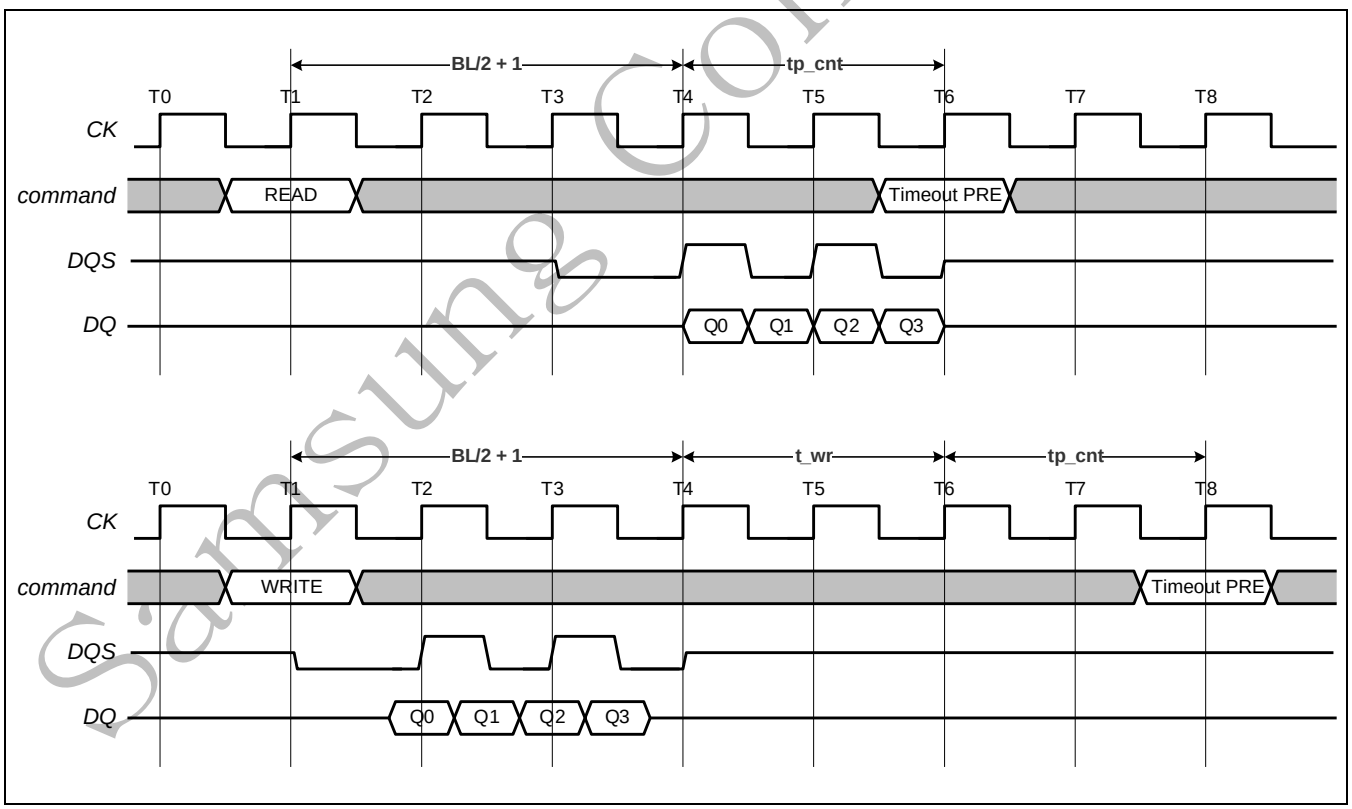

Figure 1-6 Timing Diagram Of Timeout Precharge

1.7 Quality of Service

DREX provides Quality of Service (QoS) feature to ensure low latency for real-time masters. Specifically, DREX uses timeout based QoS enforcement scheme.

- When DREX receives an AXI transaction, a predefined QoS timeout value is assigned to the corresponding memory request for timeout.
- When the timer expires, the request gets promoted to the highest priority for immediate selection during arbitration stage.

1.7.1 BRB Arbitration

The AXI requests of the head of AXI Request FIFO in the ports are arbitrated to Bank Request Buffer (BRB). The priority of each port is determined by the highest AxQoS value of the AXI requests in the FIFO. When the AXI Request FIFO timer (**ConControl.timeout_level0)** expires, the port of the FIFO has highest priority to prevent starvation of low priority port.

1.7.2 AxQoS based QoS

When DREX receives an AXI request, it assigns a predefined BRB timeout value (**QosControl_n.cfg_qos, n= 0 ~ 15**) according to the AxQoS value (0 ~ 15). When the BRB timer expires, the request has the highest priority for the Request Scheduler. When the BRB is full or the data buffer for the request is full, the BRB timer decrements by predefined value (**BRBQosConfig.brb_qos_timer_dec**) instead of 1.

1.7.3 BRB Space Reservation

When BRB space reservation is enabled for a AXI AR/AW port (**BRBRsvControl.brb_rsv_en_{w,r}{0,1,2,3}**), the AXI port stops issuing request for BRB arbitration when the occupancy of target BRB exceeds BRB threshold value (**BRBRsvConfig.brb_rsv_th_{w,r}{0,1,2,3}**).

1.7.4 Emergency priority escalation

In addition to the conventional QoS schemes, DREX-1 supports emergency signal-based priority escalation. This feature is implemented using two additional input signals:

- Each AXI port has additional ARMARKER[4:0] and AWMARKER[4:0] input signals. These signals are used for identifying they QoS class of a given request – for example, the master that sent the request, or communication that happens between two specific masters on a given application scenario.

- EMERGENCY_R[30:0] and EMERGENCY_W[30:0] input signals, which specifies which class of traffic needs a higher priority.

By default, the masters drive the value of EMERGENCY_R and EMERGENCY_W as 31'b0. When a given class of requests requires higher priority, the master drives the corresponding EMERGENCY_R (for reads) or EMERGENCY_W (for writes) bit to escalate the priority of all traffics within the given class. To be specific, when EMERGENCY_R[n] gets driven to 1, all read requests with its ARMARKER value as n+1 moves to a higher priority, and when EMERGENCY_W[n] gets driven to 1, all write requests with its AWMARKER value as n+1 moves to a higher priority.

The priority returns to its original priority as soon as the corresponding EMERGENCY_R/EMERGENCY_W signal returns to 0. Note that, the timeout counter should be still decrementing regardless of the EMERGENCY_R/EMERGENCY_W signal.

There are programmable registers (per-AxMARKER value) that determines the priority level to move to in case or priority escalation. Users can choose between the 'timeout' priority and 'urgent page miss' priority. See Section 1.14.40 and Section 1.14.41 for details of the registers.

1.8 Congestion Feedback

DREX-1 generates congestion information using the number of requests queued on the given resource. This feature is implemented two output signals IS_CONGESTED_R[p-1:0] and IS_CONGESTED_W[p-1:0], where p is the number of AXI slave ports instantiated.

To support this feature, DREX-1 has the following programmable register values per-port: (BP_CONTROLn, BP_CONFIGn_R, and BP_CONFIGn_W)

- bp_en: when this is 0, IS_CONGESTED_R and IS_CONGESTED_W is tied to 0.
- Bp_under_emergent : When this is 1, the back pressure signals are tied to 0 when EMERGENCY_R and EMERGENCY_W are 0.
- bp_off_th_data : When the read/write data buffer's availability of the port gets higher than to this threshold, IS_CONGESTED_R/IS_CONGESTED_W of the corresponding port changes to 0.
- bp_on_th_data: When the read/write data buffer's availability of the port gets lower or equal to this threshold, IS_CONGESTED_R/IS_CONGESTED_W of the corresponding port changes to 1.
- bp_off_th_brb : When the read/write BRB occupancy (maximum of all BRBs of the directions) gets higher than this threshold, IS_CONGESTED_R/IS_CONGESTED_W of the corresponding port changes to 0.
- bp_on_th_brb : When the read/write BRB occupancy (maximum of all BRBs of the directions)gets lower or equal to this threshold, IS_CONGESTED_R/IS_CONGESTED_W of the corresponding port changes to 1.

The following is the pseudo-code that describes the back pressure generating module:

```
if(bp_enable == 0) {
      is_congested_r = 0; is_congested_w = 0;
} else if(bp_under_emergent == 1 && emergency_r == 0 && emergency_w == 0) {
      is_congested_r = 0; is_congested_w = 0;
} else {
      if(read_data_buffer_level > bp_on_th_rdata) {
            read_buffer_full = 1;
      } else if(read_data_buffer_level <= bp_off_th_rdata) {
            read_buffer_full = 0;
      } else {
            read_buffer_full = (read_buffer_full from previous cycle);
      }

      If(max_read_brb_level > bp_on_th_rbrb) {
            Read_brb_full = 1;
      } else if(max_read_brb_level <= bp_off_th_rbrb) {
            Read_brb_full = 0;
      } else {
            Read_brb_full = (read_brb_full from previous cycle);
      }
```

```
        if(write_data_buffer_level > bp_on_th_wdata) {
                write_buffer_full = 1;
        } else if(write_data_buffer_level <= bp_off_th_wdata) {
                write_buffer_full = 0;
        } else {
                write_buffer_full = (write_buffer_full from previous cycle);
        }

        if(max_write_brb_level > bp_on_th_wbrb) {
                write_brb_full = 1;
        } else if(max_write_brb_level <= bp_off_th_wbrb) {
                write_brb_full = 0;
        } else {
                write_brb_full = (write_brb_full from previous cycle);
        }
        is_congested_r = read_brb_full | read_buffer_full;
        is_congested_w = write_brb_full | write_buffer_full;
}
```

1.9 Performance Profiling

DREX provides performance monitoring capability based on event counters accessible through APB interface. Relevant registers are located on 0xE000 ~ 0xE140. Note that do not access to these addres before PPCCLKCON.perev_clk_en is set to 1.

DREX has events counters which is called PPC. lists performance events in each domain.

It is possible to extract useful information from these event counts.

Table 1-1 List of Performance Events

Event Items	Clock Domain
1. Total read requests counts	aclk
2.Total read requests counts per bank	aclk
3. Total write requests counts	aclk
4.Total write requests counts per bank	aclk
6. Cas command scheduled counts	cclk
7. Page hit counts	cclk
8.DRAM data channel cycle used	cclk

Table 1-2 List of Performance Information

Information	Event Expression
Total read requests	1
Total write requests	3
Total requests	1+3
Page hit	7
Memory Data Transfer	8

For example, page_hit can be obtained like below. PPC is needed to be read.

Step1: Initialize State

Mode Selection: set perev0/1_sel to 0x68/0x6a. Then 2-bit events for PPC is {cas scheduled, page hit}.

Interrupt Enable: set INTENS_PPC(Offset=0xE030) to 0x8000_000F

Counter Enable: set CNTENS_PPC(Offset=0xE010) to 0x8000_000F

Step2: Start State

Clear Overflow Flag Register: set FLAG_PPC(Offset=0xE050) to 0x8000_000F

All Counter Reset: set PMNC_PPC(Offset=0xE000) to 0x0000_0006

Sampling Duration Initial Value Setup: set CCNT_PPC(Offset=0xE100) to some value

If 10000(0x2710) cycles simulation run, then 0xFFFF_D8EF(0xFFFF_FFFF – 0x2710) need to be set

Start All Counters: set PMNC_PPC(Offset=0xE000) to 0x0000_0001

Step3: Running & Interrupt

Interrupt will be generated from PPC due to overflow on CCNT_PPC

Step4: Stop State

Stop All Counters: set PMNC_PPC(Offset=0xE000) to 0x0000_0000

Step5: Read State

Get Overflow Flag Register: read FLAG_PPC(Offset=0xE050) value to check if interrupt has been generated by CCNT_PPC

Get Performance Counter Value: read PMCNT0~3_PPC for total cas scheduled count number and page hit count numberr.

Table 1-3 Enable/Disable & Start/Stop Control

Control Function		Description
Enable/Disable [counter/adder]		CNTENS/CNTENC
Enable/Disable [counter/adder]'s interrupt		INTENS/INTENC
Start/Stop PPC	Start_mode == 0	Start & Stop by Register : PMNC[0] == 1 → Start, PMNC[0] == 0 → Stop
	Start_mode == 1	Start & Stop by External Trigger Trigger == 1 → Start (In this case, PMNC[0] read value becomes 1) Trigger == 0 → Stop(In this case, PMNC[0] read value becomes 0)

CNTENC & CNTENS

PPMU_CTENC and PPMU_CTENS are a pair of related registers.

If you want to enable one counter, you should write 1 to its corresponding bit of PPMU_CTENS. After this writing, the values you read from PPMU_CTENC and PPMU_CTENS are both changed, which show the enabled counter's corresponding bit with 1.

If you want to disable one counter, you should write 1 to its corresponding bit of PPMU_CTENC. Still after this writing, the values you read from PPMU_CTENC and PPMU_CTENS are both changed, which show the disabled counter's corresponding bit with 0.

Their inital values are 0x0000_0000.

INTENC & INTENS

PPMU_INTENS and PPMU_INTENC are used for enabling and disabling the interrupt generation of the counters. Their setting rules are same as the two registers above.

CCNT & PMCNTx

PPMU_CCNT is an r/w register.

Before start counting, you can set PPMU_CCNT's intial value by writing some value to it. Read after this write should be the value as the same as the written value.

After starting counting (suppose that CCNT counter is enabled), the PPMU_CCNT will be increased one by every cycle from its initial value until you stop counting. Any value read or written from/to the counter registers during the counting is meaningless.

1.10 Pause Feature

DREX supports pause feature through external ports called "PAUSE_REQ" and "PAUSE_ACK" to support clock frequency change in LPDDR2-S4 and LPDDR3. Do not use this feature in DDR3.

Figure 1-7 shows the handshaking mechanism of this feature.

pause_resync_en is supported with PHY V6.

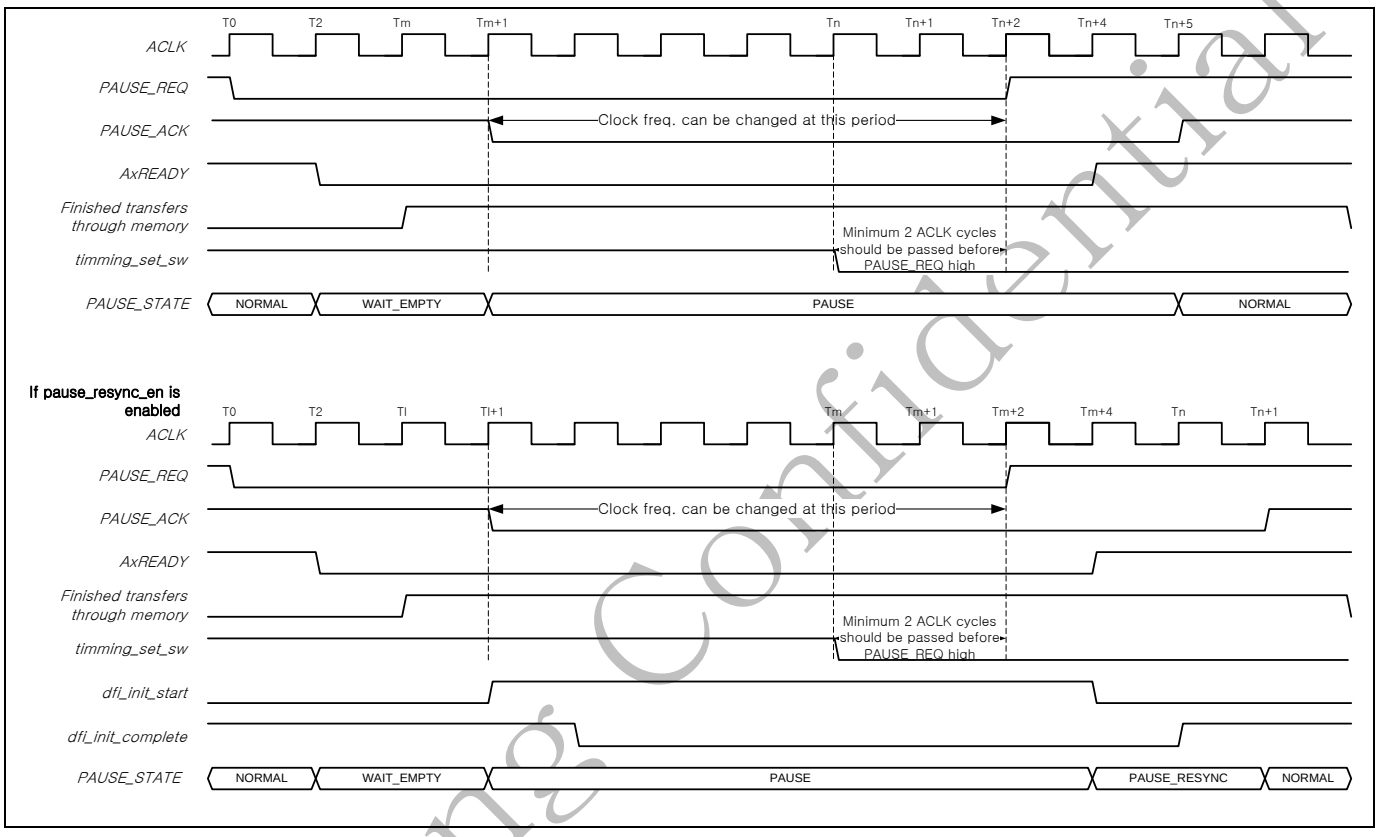

Figure 1-7 PAUSE_REQ/PAUSE_ACK handshaking

Clock frequency change of memory can be applied through this pause feature like below procedures.

1. Pause request setting PAUSE_REQ to low.
2. DREX sets AxREADY to low.
3. DREX finishes current memory access.
4. DREX sets PAUSE_ACK to low.
5. Toggle the timing set switch for proper value by external port(timing_set_sw) then change the clock frequency.
6. Release pause request setting PAUSE_REQ to high.
7. DREX sets AxREADY and PAUSE_ACK to high.

It is recommended that toggling the timing set switch before clock frequency change to obtain the switching time safely. The minimum switching time is 2 ACLK cycles for switching by external port.

Master side of this protocol should not change PAUSE_REQ value before finishing previous handshaking. It means that PAUSE_REQ should be waiting for the PAUSE_ACK before change its value.
DREX does not support "Not Acknowledged" in DFI frequency change protocol.

Although there are two timing parameter register sets which can be switched for high speed and low speed operation, below timing parameter should have the same value in register set 0 and 1.

- TimingData.t_w2w_c2c, t_r2r_c2c, wl, rl

If port_policy (in PrechConfig0.port_policy) is enabled, nWR value in LPDDR2-S4/LPDDR3 MR1 can be conflicted with operation frequency. Be careful to use auto precharge mode in frequency change scenario.

1.11 Trust Zone Address Space Control (TZASC)

TZASC performs security checks on AXI accesses to memory. This supports configurable number of regions. Each region is programmable for size, base address, enable, and security parameters. Using the secure_boot_lock input signal, the programmers view can be locked to prevent erroneous writes. It provides programmability in reporting faults using AXI response channel, and interrupt.

1.11.1 Regions

A region is a contiguous area of address space. The TZASC provides each region with a programmable security permissions field. The security permissions value is used to enable the TZASC to either accept or deny a transaction access to that region. The transactions arprots[2:0] or awprots[2:0] signals are used to determine the security settings of that transaction. The region features are as following:

- Minimum region granularity: 64kB
- Number of regions: 8 non-overlapping regions (region 1~8) + 1 default region (region 0)
- Priority of regions: priority_of_region_1-8 > priority_of_region_0
- Address space of a region:
  - Base address: aligned to 64kB
  - Size of a region: 64kB x N
  - Maximum size of a region: 4GB
  - Region 0 covers entire address space
- No support of subregions

The feature of security property of regions are as following:

- 4 bits permission
  - Each bit represent if (secure read, secure write, non-secure read, non-secure write) are allowed or not
  - EX) A region with security property b'1100 allows secure access (read and write) but forbids non-secure access
- 1 bit region_lock
  - Region configuration (SFR) is locked after secure_boot_lock is asserted
- 1bit region enable
  - Region 1~8 is enabled by SFR
  - Region 0 is always enabled.

1.11.2 Denied AXI transactions

If an AXI transaction has insufficient security privileges then for:

- Reads: The TZASC responds to the master by setting all bits of the read data bus, RDATA, to zero.
- Writes: The TZASC prevents the data from being written to memory.

The TZASC Action Register controls whether the TZASC signals to the master when a region permission failure occurs, and if so, the type of response it provides.

1.11.3 Preventing writes to registers and using secure_boot_lock

The TZASC expects the secure_boot_lock signal to be asserted for at least one clock cycle. One clock after the secure_boot_lock is sampled HIGH by TZASC, then the below registers cannot be written, unless the DREX is reset by asserting ARESETn.

Assergin secure_boot_lock signal makes the following register read only:

- TZASC Lockdown Select Register

By suitably programming TZASC Lockdown Select Register and asserting secure_boot_lock signal makes the following register read only:

- TZASC Lockdown Range Register

By programming the TZASC Lockdown Select Register, and TZASC Lockdown Range Register, and asserting the secure_boot_lock signal, you can lockdown the behavior of the TZASC so that it prevents unintentional or erroneous write to the regions specified in the TZASC Lockdown Range Register. However, read access to those regions is permitted:

- TZASC Region Setup Low Register n
- TZASC Region Setup High Register n
- TZASC Region Attribute Register n

1.11.4 Using exclusive accesses

If a master performs exclusive accesses to an address region, you must program the TZASC to permit read and write accesses to that address region, for the expected settings of arprots[1] and awprots[1], otherwise the read or write transaction might fail. The TZASC permission failed exclusive accesses do not modify the state of exclusive monitor.

1.12 Data Buffer Configuration

Data buffer resources for each port is configurable. This feature can be used for port which has none performance critical properties to reduce the area. When instantiating, use parameter overriding method to override the value.

The parameter name is as below.

PAR_RBUF_ENTRIES_P0/1/2/3 represent the number of entries on the read reordering queue which is used for tracking the number of AR requests that the specific port can track and configured to 4/8/16.

- parameter PAR_RBUF_ENTRIES_P0 = 16;
- parameter PAR_RBUF_ENTRIES_P1 = 16;
- parameter PAR_RBUF_ENTRIES_P2 = 16;
- parameter PAR_RBUF_ENTRIES_P3 = 16;

PAR_R(W)BUF_SIZE_P0/1/2/3 represent the number of data buffer entries and configured to 4/5/6 which means that the number of data buffer entries is 16/32/64 repectively.

- parameter PAR_R(W)BUF_SIZE_P0 = 6;
- parameter PAR_R(W)BUF_SIZE_P1 = 6;
- parameter PAR_R(W)BUF_SIZE_P2 = 6;
- parameter PAR_R(W)BUF_SIZE_P3 = 6;

1.13 Clock Gating

DREX has clock gating feature for internal logic and PHY clock.

- Internal logic clock gating enable: int_cg_en in ConControl register
- PHY clock gating enable: phy_cg_en in ConControl register

Figure 1-8 shows block diagram of DREX clock gating.

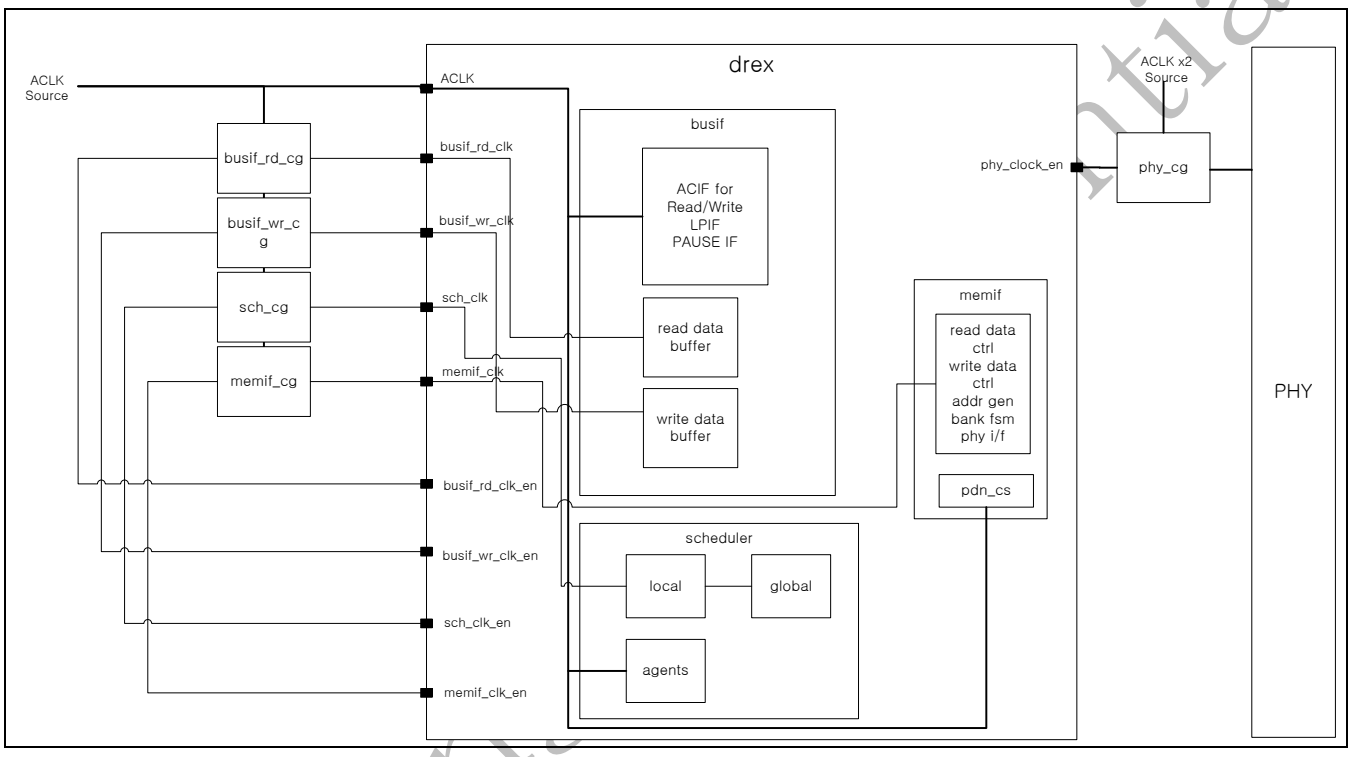

Figure 1-8 Clock Gating Block Diagram

Timing diagram for phy_clock_en and gated phy clock (2x clock of ACLK) is illustrated in Figure 1-9. From phy_clock_en to phy_cg input can be set to multicycle path.

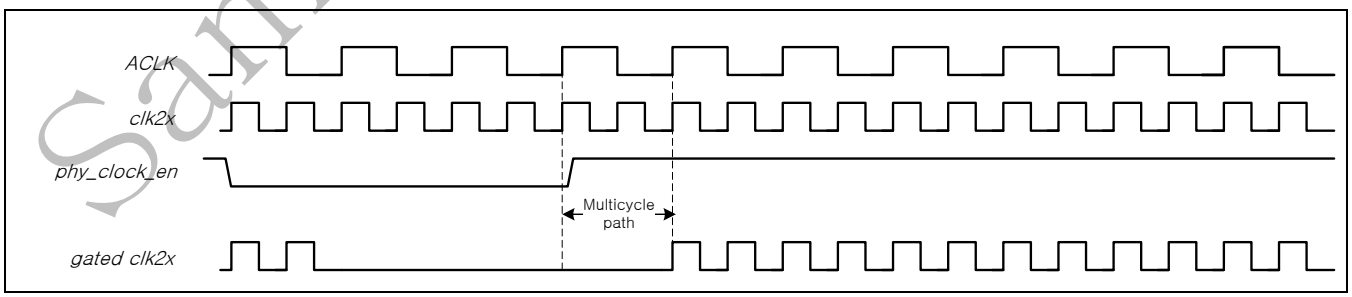

Figure 1-9 Timing Diagram from phy_clock_en to phy_cg input

1.14 Register Descriptions

1.14.1 Register Overview

Register	Offset	R/W	Description	Initial Value
General Registers				
CONCONTROL	0x0000	R/W	Controller Control Register	0x0FFF_1100
MEMCONTROL	0x0004	R/W	Memory Control Register	0x0020_2601
CGCONTROL	0x0008	R/W	Clock Gating Control Register	0x0000_0000
DIRECTCMD	0x0010	R/W	Memory Direct Command Register	0x0000_0000
PRECHCONFIG0	0x0014	R/W	Precharge Policy Configuration0 Register	0x0000_0000
PHYCONTROL0	0x0018	R/W	PHY Control0 Register	0x0000_0000
PRECHCONFIG1	0x001C	R/W	Precharge Policy Configuration1 Register	0xFFFF_FFFF
TIMINGRFCPB	0x0020	R/W	AC Timing Register for SDRAM Per Bank Refresh	0x0000_1818
PWRDNCONFIG	0x0028	R/W	Dynamic Power Down Configuration Register	0xFFFF_00FF
TIMINGPZQ	0x002C	R/W	AC Timing Register for DDR3 SDRAM periodic ZQ(ZQCS)	0x0000_4084
TIMINGAREF	0x0030	R/W	AC Timing Register for SDRAM Auto refresh	0x0005_005D
TIMINGROW0	0x0034	R/W	AC Timing Register0 for SDRAM Row	0x1F23_3286
TIMINGDATA0	0x0038	R/W	AC Timing Register0 for SDRAM Data	0x1230_065C
TIMINGPOWER0	0x003C	R/W	AC Timing Register0 for Power Mode of SDRAM	0x381B_0422
PHYSTATUS	0x0040	R	PHY Status Register	0x0000_0000
ETCTIMING	0x0044	R/W	AC Timing for WideIO and Other Parameters	0x0000_2270
CHIPSTATUS	0x0048	R	Memory Status Register	0x0000_0000
MRSTATUS	0x0054	R	Memory Mode Registers Status Register	0x0000_0000
QOSCONTROL0	0x0060	R/W	Quality of Service Control Register 0	0x0000_0FFF
QOSCONTROL1	0x0068	R/W	Quality of Service Control Register 1	0x0000_0FFF
QOSCONTROL2	0x0070	R/W	Quality of Service Control Register 2	0x0000_0FFF
QOSCONTROL3	0x0078	R/W	Quality of Service Control Register 3	0x0000_0FFF
QOSCONTROL4	0x0080	R/W	Quality of Service Control Register 4	0x0000_0FFF
QOSCONTROL5	0x0088	R/W	Quality of Service Control Register 5	0x0000_0FFF
QOSCONTROL6	0x0090	R/W	Quality of Service Control Register 6	0x0000_0FFF
QOSCONTROL7	0x0098	R/W	Quality of Service Control Register 7	0x0000_0FFF
QOSCONTROL8	0x00A0	R/W	Quality of Service Control Register 8	0x0000_0FFF
QOSCONTROL9	0x00A8	R/W	Quality of Service Control Register 9	0x0000_0FFF
QOSCONTROL10	0x00B0	R/W	Quality of Service Control Register 10	0x0000_0FFF
QOSCONTROL11	0x00B8	R/W	Quality of Service Control Register 11	0x0000_0FFF

QOSCONTROL12	0x00C0	R/W	Quality of Service Control Register 12	0x0000_0FFF
QOSCONTROL13	0x00C8	R/W	Quality of Service Control Register 13	0x0000_0FFF
QOSCONTROL14	0x00D0	R/W	Quality of Service Control Register 14	0x0000_0FFF
QOSCONTROL15	0x00D8	R/W	Quality of Service Control Register 15	0x0000_0FFF
TIMING_SET_SW	0x00E0	R/W	Quality of Service Control Register 15	0x0000_0000
TIMINGROW1	0x00E4	R/W	AC Timing Register1 for SDRAM Row	0x1F23_3286
TIMINGDATA1	0x00E8	R/W	AC Timing Register1 for SDRAM Data	0x1238_367C
TIMINGPOWER1	0x00EC	R/W	AC Timing Register1 for Power Mode of SDRAM	0x381B_0422
WRTRA_CONFIG	0x00F4	R/W	Write Training Configuration Register	0x0000_0000
RDLVL_CONFIG	0x00F8	R/W	Read Leveling Configuration Register(For PHY V5 only)	0x0000_0000
PEREVCONFIG	0x00FC	R/W	Performance Events Configuration Register	0x0000_0000
BRBRSVCONTROL	0x0100	R/W	BRB Reservation Control Register	0x0000_0000
BRBRSVCONFIG	0x0104	R/W	BRB Reservation Configuration Register	0x8888_8888
BRBQOSCONFIG	0x0108	R/W	BRB QoS Configuration Register	0x0000_0010
WRLVL_CONFIG0	0x0120	R/W	Write Leveling Configuration Register 0	0x0000_0010
WRLVL_CONFIG1	0x0124	R/W	Write Leveling Configuration Register 1	0x0000_0000
WRLVL_STATUS	0x0128	R	Write Leveling Status Register	0x0000_0001
PPCCLKCON	0x0130	R/W	Performance Event Clock Control Register	0x0000_0000
PEREVCONFIG0	0x0134	R/W	Performance Event Configuration0 Register	0x0000_0000
PEREVCONFIG1	0x0138	R/W	Performance Event Configuration1 Register	0x0000_0000
PEREVCONFIG2	0x013C	R/W	Performance Event Configuration2 Register	0x0000_0000
PEREVCONFIG3	0x0140	R/W	Performance Event Configuration3 Register	0x0000_0000
CTRL_IO_RDATA	0x0150	R	CTRL_IO_RDATA from PHY	0x0000_0000
CACAL_CONFIG0	0x0160	R/W	CA Calibration Configuration Register 0	0x003F_F010
CACAL_CONFIG1	0x0164	R/W	CA Calibration Configuration Register 1	0x0000_0000
CACAL_STATUS	0x0164	R	CA Calibration Status Register	0x0000_0000
EMERGENT_CONFIG0	0x0200	R/W	Emergent Configuration Registe 0	0x0000_0000
EMERGENT_CONFIG1	0x0204	R/W	Emergent Configuration Registe 1	0x0000_0000
BP_CONTROL0	0x0210	R/W	Back Pressure Control Register 0	0x0000_0002
BP_CONFIG0_R	0x0214	R/W	Back Pressure Configuration Register 0 for Read	0x0000_0000
BP_CONFIG0_W	0x0218	R/W	Back Pressure Configuration Register 0 for Write	0x0000_0000
BP_CONTROL1	0x0220	R/W	Back Pressure Control Register 1	0x0000_0002
BP_CONFIG1_R	0x0224	R/W	Back Pressure Configuration Register 1 for Read	0x0000_0000
BP_CONFIG1_W	0x0228	R/W	Back Pressure Configuration Register 1 for Write	0x0000_0000
BP_CONTROL2	0x0230	R/W	Back Pressure Control Register 2	0x0000_0002
BP_CONFIG2_R	0x0234	R/W	Back Pressure Configuration Register 2 for Read	0x0000_0000

BP_CONFIG2_W	0x0238	R/W	Back Pressure Configuration Register 2 for Write	0x0000_0000
BP_CONTROL3	0x0240	R/W	Back Pressure Control Register 3	0x0000_0002
BP_CONFIG3_R	0x0244	R/W	Back Pressure Configuration Register 3 for Read	0x0000_0000
BP_CONFIG3_W	0x0248	R/W	Back Pressure Configuration Register 3 for Write	0x0000_0000
WINCONFIG_ODT_W	0x0300	R/W	Window Configuration Register for Wirte ODT	0x0000_0000
WINCONFIG_CTRLREAD	0x0308	R/W	Window Configuration Register for CTRLREAD	0x0000_0000
WINCONFIG_CTRLGATE	0x030C	R/W	Window Configuration Register for CTRLGATE	0x0000_0000
PMNC_PPC	0xE000	R/W	Performance Monitor Control Register	0x0000_0000
CNTENS_PPC	0xE010	R/W	Count Enable Set Register	0x0000_0000
CNTENC_PPC	0xE020	R/W	Count Enable Clear Register	0x0000_0000
INTENS_PPC	0xE030	R/W	Interrupt Enable Set Register	0x0000_0000
INTENC_PPC	0xE040	R/W	Interrupt Enable Clear Register	0x0000_0000
FLAG_PPC	0xE050	R/W	Overflow Flag Status Register	0x0000_0000
CCNT_PPC	0xE100	R/W	Cycle Count Register	0x0000_0000
PMCNT0_PPC	0xE110	R/W	Performance Monitor Count Register	0x0000_0000
PMCNT1_PPC	0xE120	R/W	Performance Monitor Count Register	0x0000_0000
PMCNT2_PPC	0xE130	R/W	Performance Monitor Count Register	0x0000_0000
PMCNT3_PPC	0xE140	R/W	Performance Monitor Count Register	0x0000_0000

1.14.2 Controller Control Register (ConControl, R/W, Address Offset=0x0000)

Field	Bit	Description	R/W	Initial State
Reserved	[31:29]	Should be zero		0x0
dfi_init_start	[28]	DFI PHY initialization start This field is used to initialize DFI PHY. Set this field to 1 to initialize DFI PHY and set this field to 0 after received dfi_init_complete of PhyStatus register.	R/W	0x0
timeout_level0	[27:16]	**Default Timeout Cycles** 0xn = n aclk cycles (aclk : AXI clock) This counter prevents transactions in the AXI request FIFO from starvation. This counter starts if a new AXI transaction comes into the request FIFO. If the counter becomes zero, the corresponding FIFO has the highest priority during BRB arbitration. Refer to chapter 1.7 . Quality of Service for detailed information	R/W	0xFFF
Reserved	[15]	Should be zero		0x0
rd_fetch	[14:12]	**Read Data Fetch Cycles** 0xn = n cclk cycles (cclk : DREX core clock) The recommended value of this field is 0x2 for LPDDR3 800Mhz memory clock and other cases are 0x1. This register is for the unpredictable latency of read data coming from memory devices by tDQSCK variation or the board flying time. The read fetch delay of PHY read FIFO must be controlled by this parameter. The controller will fetch read data from PHY after read_latency + n cclk cycles. In Wide IO Memory, the recommended value is 1.	R/W	0x1
Reserved	[11:9]	Should be zero		0x0
empty	[8]	**Empty Status** 0x0 = Not Empty, **0x1 = Empty** There is no AXI transaction in memory controller.	R	0x1
io_pd_con	[7:6]	**I/O Receiver Powerdown Control** 0x0 = Use programmed ctrl_pd, 0x1 = Automatic control for ctrl_pd in CKE low, 0x2 = Automatic control for ctrl_pd in none read state, 0x3 = Reserved "ctrl_pd" is used for powerdown of I/O cell receiver. If this field is set to 0x0, DREX only sends programmed ctrl_pd value. If this field is set to 0x1 and memory state is in CKE low state, DREX automatically set powerdown enable for input buffer of I/O. If this value is set to 0x2 and memory state is in none read,	R/W	0x0

		DREX automatically set powerdown enable for input buffer of I/O. Note that this field should be turn off during initialization and training sequence.		
aref_en	[5]	**Auto Refresh Counter** 0x0 = Disable, 0x1 = Enable Enable this to decrease the auto refresh counter by 1 at the rising edge of the rclk Note that this field should be turn off during initialization and training sequence.	R/W	0x0
Reserved	[4]	Should be zero		0x0
update_mode	[3]	**The kind of Update Interface in DFI** 0x0 = PHY initiated update/acknowledge mode 0x1 = MC initiated update/acknowledge mode In case of wide io memory and PHY V5, this field should be 1'b1.	R/W	0x0
clk_ratio	[2:1]	**Clock Ratio of Bus Clock to Memory Clock** 0x0 = freq.(aclk): freq.(cclk) = 1: 1, 0x1 ~ 0x3 = Reserved	R/W	0x0
ca_swap	[0]	**CA Swap for LPDDR2-S4/LPDDR3** 0x0 = CA swap disable, 0x1 = CA swap enable If this field is enabled, ca[9:0] is reversed to ca[0:9].	R/W	0x0

1.14.3 Memory Control Register (MemControl, R/W, Address Offset=0x0004)

Field	Bit	Description	R/W	Initial State
Reserved	[31:30]	Should be zero		0x0
pause_ref_en	[29]	**Refresh Command Issue Before PAUSE ACKNOLEDGE** 0x0 = Disable, 0x1 = Enable If this field is enabled, refresh command is issued before PAUSE acknowledge. If pzq_en is enabled, do not enable this field.	R/W	0x0
sp_en	[28]	**Read with Short Preamble in Wide IO Memory** 0x0 = Disable, 0x1 = Enable Wide IO Memory has read with short preamble command to support zero-bubble in chip to chip read cyles. If this field is enabled, t_r2r_c2c is neglected.	R/W	0x0
pb_ref_en	[27]	**Per Bank Refresh for LPDDR2-S4/LPDDR3** 0x0 = Disable, 0x1 = Enable To use per bank refresh feature, turn on this field before aref_en is enabled in ConControl register. Per bank refresh is only allowed in LPDDR2-S4 and LPDDR3 devices with 8 banks and PHY update mode (ConControl.update_mode = 0). PHY update mode is only supported with PHY V6. So this feature is not supported with PHY V5.	R/W	0x0
Reserved	[26:25]	Should be zero		0x0
pzq_en	[24]	DDR3 periodic ZQ(ZQCS) enable Note that after exit from self refresh, ZQ function is required by the software. The controller does not support ZQ calibration command to be issued in parallel to DLL lock time when coming out of self refresh. Turn-on only when using DDR3. The periodic ZQ interval is defined by t_pzq in TIMINGPZQ register.	R/W	0x0
bl	[22:20]	**Memory Burst Length** 0x0 = Reserved, 0x1 = 2 (Wide IO Memory use only), 0x2 = 4, 0x3 = 8, 0x4 ~ 0x7 = Reserved In case of Wide IO Memory, the controller only supports burst length 2. In case of LPDDR2-S4, the controller only supports burst length 4. In case of DDR3 and LPDDR3, the controller only supports burst length 8.	R/W	0x2
num_chip	[19:16]	**Number of Memory Chips**	R/W	0x0

		0x0 = 1 chip, 0x1 = 2 chips, 0x2 ~ 0xf = Reserved		
mem_width	[15:12]	**Width of Memory Data Bus** 0x0 ~ 0x1 = Reserved, 0x2 = 32-bit, 0x3 = Reserved 0x4 = 128-bit (Wide IO Memory use only), 0x5 ~ 0xf = Reserved	R/W	0x2
mem_type	[11:8]	Type of Memory 0x0 = Wide IO Memory, 0x1 ~ 0x4 = Reserved, 0x5 = LPDDR2-S4, 0x6 = DDR3, 0x7 = LPDDR3, 0x8 ~ 0xf = Reserved	R/W	0x6
add_lat_pall	[7:6]	**Additional Latency for PALL in cclk cycle** 0x0 = 0 cycle, 0x1 = 1 cycle 0x2 = 2 cycle, 0x3 = Reserved If all banks precharge command is issued, the latency of pre-charging will be tRP + add_lat_pall	R/W	0x0
dsref_en	[5]	**Dynamic Self Refresh** 0x0 = Disable, 0x1 = Enable Refer to chapter 1.5.2 . Dynamic power down for detailed information. In DDR3, this feature is not supported.	R/W	0x0
Reserved	[4]	Should be zero		0x0
dpwrdn_type	[3:2]	**Type of Dynamic Power Down** 0x0 = Active/precharge power down, 0x1 = Forced precharge power down 0x2 ~ 0x3 = Reserved Refer to chapter 1.5.2 . Dynamic power down for detailed information.	R/W	0x0
dpwrdn_en	[1]	**Dynamic Power Down** 0x0 = Disable, 0x1 = Enable	R/W	0x0
clk_stop_en	[0]	**Dynamic Clock Control** 0x0 = Always running, 0x1 = Stops during idle periods	R/W	0x1

		This feature is only supported with LPDDR2/LPDDR3. Refer to chapter 1.5.4 . Clock stop for detailed information. Note that if phy_cg_en is enabled then this field should be zero.		

1.14.4 Clock Gating Control Register (CGControl, R/W, Address Offset=0x0008)

This register should not be written to enable after all initialization and trainings have been finished.

Field	Bit	Description	R/W	Initial State
Reserved	[31:5]	Should be zero		0x0
phy_cg_en	[4]	**PHY Clock Gating** 0x0 = phy clock gating disable 0x1 = phy clock gating enable Note that if this field is enabled then clk_stop_en(MemControl) should be set to zero. This feature is not supported in DDR3 (this field should be zero in case of DDR3.)	R/W	0x0
me-mif_cg_en	[3]	**Memory Controller Internal Clock Gating - Memory I/F** 0x0 = Memory i/f clock gating disable 0x1 = Memory i/f clock gating enable	R/W	0x0
scg_cg_en	[2]	**Memory Controller Internal Clock Gating - Scheduler** 0x0 = Scheduler clock gating disable 0x1 = Scheduler clock gating enable	R/W	0x0
bu-sif_wr_cg_en	[1]	**Memory Controller Internal Clock Gating - BUS I/F Write** 0x0 = bus i/f write clock gating disable 0x1 = bus i/f write clock gating enable	R/W	0x0
bu-sif_rd_cg_en	[0]	**Memory Controller Internal Clock Gating - BUS I/F Read** 0x0 = bus i/f read clock gating disable 0x1 = bus i/f read clock gating enable	R/W	0x0

1.14.5 Memory Direct Command Register (DirectCmd, R/W, Address Offset=0x0010)

Field	Bit	Description	R/W	Initial State
Reserved	[31:28]	Should be zero.		0x0
cmd_type	[27:24]	**Type of Direct Command** 0x0 = MRS/EMRS (mode register setting), 0x1 = PALL (all banks precharge), 0x2 = PRE (per bank precharge), 0x3 = Reserved, 0x4 = REFS (self refresh), 0x5 = REFA (auto refresh), Do not use this command if pb_ref_en is enabled in MemControl register. 0x6 = CKEL (active/precharge power down), 0x7 = NOP (exit from active/precharge power down, 0x8 = REFSX (exit from self refresh), 0x9 = MRR (mode register reading), 0xa = ZQINIT(ZQ calibration init.) 0xb = ZQOPER(ZQ calibration long) 0xc = ZQCS(ZQ calibration short) 0xd = SRR for Wide IO Memory 0xe ~ 0xf = Reserved When a direct command is issued, AXI masters must not access memory. It is strongly recommended to check the command queue's state by ConControl.chip0/1_empty and the chip FSM in the ChipStatus register before issuing a direct command. The chip status must be checked before issuing a direct command. And clk_stop_en, dynamic power down, dynamic self refresh, force precharge function (MemControl register) and sl_dll_dyn_con (PhyControl0 register) must be disabled. MRS/EMRS or MRR commands should be issued if all banks are in idle state. If MRS/EMRS or MRR is issued to LPDDR2-S4/LPDDR3, the CA pins must be mapped as follows. MA[7:0] = {cmd_addr[1:0], cmd_bank[2:0], cmd_addr[12:10]}, OP[7:0] = cmd_addr[9:2] In DDR3, self refresh related timing such as tCKESR/tCKSRE/tCKSRX should be check by software. Note that do not write reserved value to this field. In DPD command, if two chip is used and DPD command is issued to certain chip, then some illegal command or odt can be occurred to DPD chip by dynamic powerdown control or termination control.	R/W	0x0
Reserved	[23:21]	Should be zero.		0x0
cmd_chip	[20]	**Chip Number to send the direct command to**	R/W	0x0

		0 = Chip 0 1 = Chip 1		
cmd_addr_16	[19]	**A16 for WideIO Memory** A16 bit in WideIO Memory mode register definition		0x0
cmd_bank	[18:16]	**Related Bank Address when issuing a direct command** To send a direct command to a chip, additional information such as the bank address is required. This register is used in such situations.	R/W	0x0
cmd_addr	[15:0]	**Related Address value when issuing a direct command** To send a direct command to a chip, additional information such as the address is required. This register is used in such situations.	R/W	0x0

1.14.6 Precharge Policy Configuration0 Register (PrechConfig0, R/W, Address Offset=0x14)

Field	Bit	Description	R/W	Initial State
tp_en	[31:28]	**Timeout Precharge per Port** 0x0 = Disable, 0x1 = Enable [31:28] is timeout precharge enable bit for port0, 1, 2, 3 repectively. If tp_en is enabled, it automatically precharges an open bank after a specified amount of mclk cycles (if no access has been made in between the cycles) in an open page policy. If **PrechConfig1.tp_cnt** bit-field is set, it specifies the amount of mclk cycles to wait until timeout precharge precharges the open bank. Refer to chapter 1.6.2 . Timeout precharge for detailed information.	R/W	0x0
Reserved	[27:20]	Should be zero		0x0
port_policy	[19:16]	**Memory Precharge Port Selective Policy** 0x0 = Open page policy, 0x1 = Close page (auto precharge) policy port_policy[n], n is the port number.	R/W	0x0
Reserved	[15:0]	Should be zero		0x0

1.14.7 PHY Control0 Register (PhyControl0, R/W, Address Offset=0x0018)

Field	Bit	Description	R/W	Initial State
mem_term_en	[31]	**Termination Enable for Memory Write ODT** At high-speed memory operation, it can be necessary to turn on termination resistance in memory devices. This register controls an ODT pin for write of memory device. All ODT pins will be high when any chip is in write state. This field is only applicable in LPDDR3/DDR3 like below. - LPDDR3 800Mhz (RL 12, WL 6) - DDR3 800Mhz In LPDDR3-800Mhz(RL 12, WL6), if mem_term_en or phy_term_en is enabled, dqsck in TimingData0/1 register need be set to bigger value than RU(tDQSCK max/tCK).	R/W	0x0
phy_term_en	[30]	**Termination Enable for PHY** At high-speed memory operation, it can be necessary to turn on termination resistance in the PHY. In LPDDR3-800Mhz(RL 12, WL6), if mem_term_en or phy_term_en is enabled, dqsck in TimingData0/1 register need be set to bigger value than RU(tDQSCK max/tCK).	R/W	0x0
ctrl_shgate	[29]	**Duration of DQS Gating Signal** This field controls the gate control signal In LPDDR2-S4/LPDDRD3, this field should be 1'b0 regardless of clock frequency. In DDR3, according to memory clock, set the value like below. 1'b0 = (gate signal length = "burst length / 2" (<= 200MHz)) 1'b1 = (gate signal length = "burst length / 2" – 1 (> 200MHz)) In Wide IO memory, set this value to 1.		0x0
ctrl_pd	[28:24]	**Input Gate for Power Down** If this field is set, input buffer is off for power down. This field should be 0 for normal operation. ctrl_pd[4:0] = for each data slice 4..0	R/W	0x0
Reserved	[23:9]	Should be zero		0x0
mem_term_type	[8]	**Termination Type for Memory Write ODT** 1'b0 = enable only single chip ODT of write operation 1'b1 = enable both chip ODT during write operation	R/W	0x0
pause_resync_en	[7]	**Resync Enable During PAUSE Handshaking** This field is to enable PHY resync before exiting PAUSE handshaking. Refer to 1.10 . 0 = Disable 1 = Enable This field should not be changed during PAUSE handshaking. Note that this field is not valid in PHY V5 which means that	R/W	0

		this field should be off with PHY V5.		
dqs_delay	[6:4]	**Delay Cycles for DQS Cleaning** This register is to enable PHY to clean incoming DQS signals delayed by external circumstances. If DQS is coming with read latency plus n memory clock cycles, this registers must be set to n memory clock cycles. In Wide IO memory, the recommended value is 1. In other type of memory, the recommended value is 0.	R/W	0x0
fp_resync	[3]	**Force DLL Resyncronization**	R/W	0x0
drv_bus_en	[2]	**Drive Memory DQ Bus signals** 0 = Disable 1 = Enable If this field is enabled, ctrl_drv_bus port will be high during memory dq bus is idle		0x0
sl_dll_dyn_con	[1]	**Turn On PHY Slave DLL Dynamically** 0 = Disable 1 = Enable	R/W	0x0
mem_term_chips	[0]	**Memory Termination Between Chips for Read** 0 = Disable 1 = Enable This field is only valid when num_chip is 0x1(2 chips) in MemControl register and DDR3. This register controls an ODT pin for read of memory device. ODT pin except read state chip will be high.	R/W	0x0

1.14.8 Precharge Policy Configuration1 Register (PrechConfig1, R/W, Address Offset=0x1C)

Field	Bit	Description	R/W	Initial State
tp_cnt3	[31:24]	**Timeout Precharge Cycles** 0xn = n cclk cycles, The minimum vlaue of this field is 0x2	R/W	0xFF
tp_cnt2	[31:24]	**Timeout Precharge Cycles** 0xn = n cclk cycles, The minimum vlaue of this field is 0x2	R/W	0xFF
tp_cnt1	[31:24]	**Timeout Precharge Cycles** 0xn = n cclk cycles, The minimum vlaue of this field is 0x2	R/W	0xFF
tp_cnt0	[7:0]	**Timeout Precharge Cycles** 0xn = n cclk cycles, The minimum vlaue of this field is 0x2 If the timeout precharge function (MemControl.tp_en) is enabled and the timeout precharge counter becomes zero, the controller forces the activated memory bank into the pre-charged state. Refer to chapter 1.6.2 .Timeout precharge for detailed information.	R/W	0xFF

1.14.9 AC Timing Register for Per Bank Refresh of Memory (TimingRFCpb, R/W, Address Offset=0x0020)

Field	Bit	Description	R/W	Initial State
Reserved	[31:14]	Should be zero		0x0
t_rfcpb1	[13:8]	Per Bank Auto refresh to Active / Per Bank Auto refresh for command period, in cclk cycles for timing set #1 t_rfcpb * T(cclk) should be greater than or equal to the minimum value of memory tRFCpb. The minimum value is 3.	R/W	0x18
Reserved	[7:6]	Should be zero		0x0
t_rfcpb0	[5:0]	Per Bank Auto refresh to Active / Per Bank Auto refresh command period, in cclk cycles for timing set #0 t_rfcpb * T(cclk) should be greater than or equal to the minimum value of memory tRFCpb. The minimum value is 3.	R/W	0x18

1.14.10 Dynamic Power Down Configuration Register (PwrdnConfig, R/W, Address Offset=0x0028)

Field	Bit	Description	R/W	Initial State
dsref_cyc	[31:16]	**Number of Cycles for dynamic self refresh entry** 0xn = n aclk cycles, The minimum value of this field is 0x2.	R/W	0xFFFF

		If the command queue is empty for n+1 cycles, the controller forces memory devices into self refresh state. Refer to chapter 1.5.3 . Dynamic self refresh for detailed information.		
Reserved	[15:8]	Should be zero		0x0
dpwrdn_cyc	[7:0]	**Number of Cycles for dynamic power down entry** 0xn = n aclk cycles, The minimum value of this field is 0x2. If the command queue is empty for n+1 cycles, the controller forces the memory device into active/precharge power down state. Refer to chapter 1.5.2 . Dynamic power down for detailed information.	R/W	0xFF

1.14.11 AC Timing Register for Periodic ZQ(ZQCS) of Memory (TimingPZQ, R/W, Address Offset=0x002C)

Field	Bit	Description	R/W	Initial State
Reserved	[31:24]	Should be zero		0x0
t_pzq	[23:0]	Average Periodic ZQ Interval(Only in DDR3) tREFI(t_refi * T(rclk)) * t_pzq should be less than or equal to the minimum value of memory periodic ZQ interval, for example, if rclk frequency is 12MHz, t_refi is set to 93 and ZQ interval is 128ms then the following value should be programmed into it: 128 ms * 12 MHz / 93 = 16516 The minimum value is 2.		0x4084

1.14.12 AC Timing Register for Auto Refresh of Memory (TimingAref, R/W, Address Offset=0x0030)

Field	Bit	Description	R/W	Initial State
t_refipb	[31:16]	Average Periodic Refresh Interval t_refipb * T(rclk) should be less than or equal to the minimum value of memory tREFIpb (per bank), for example, for the all bank refresh period of 0.4875us, and an rclk frequency of 12MHz, the following value should be programmed into it: 0.4875 us * 12 MHz = 5	R/W	0x5
t_refi	[15:0]	Average Periodic Refresh Interval t_refi * T(rclk) should be less than or equal to the minimum value of memory tREFI (all bank), for example, for the all bank refresh period of 7.8us, and an rclk frequency of 12MHz, the following value should be programmed into it: 7.8 us * 12 MHz = 93	R/W	0x5D

1.14.13 AC Timing Register n for the Row of Memory (TimingRow n, R/W, Address Offset=0x0034(for n = 0), 0x00E4(for n = 1))

Field	Bit	Description	R/W	Initial State
t_rfc	[31:24]	All Bank Auto refresh to Active / All Bank Auto refresh command period, in cclk cycles t_rfc * T(cclk) should be greater than or equal to the minimum value of memory tRFC. The minimum value is 17 if PHY is running with dll on. In FPGA with low frequency and dll is off, the minimum value is 3.	R/W	0x1F
t_rrd	[23:20]	Active bank A to Active bank B delay, in cclk cycles t_rrd * T(cclk) should be greater than or equal to the minimum value of memory tRRD.	R/W	0x2

		The minimum value is 2.		
t_rp	[19:16]	Precharge command period, in cclk cycles t_rp * T(cclk) should be greater than or equal to the minimum value of memory tRP. The minimum value is 2.	R/W	0x3
t_rcd	[15:12]	Active to Read or Write delay, in cclk cycles t_rcd * T(cclk) should be greater than or equal to the minimum value of memory tRCD + T(cclk)/2. For example, tRCD in memory specification is 13.75ns and cclk is 3.0ns, t_rcd * 3ns >= 13.75ns + 1.5ns The right value for t_rcd is 6. The minimum value is 2.	R/W	0x3
t_rc	[11:6]	Active to Active period, in cclk cycles t_rc * T(cclk) should be greater than or equal to the minimum value of memory tRC. The minimum value is 2.	R/W	0xA
t_ras	[5:0]	Active to Precharge command period, in cclk cycles t_ras * T(cclk) should be greater than or equal to the minimum value of memory tRAS + T(cclk)/2. For example, tRAS in memory specification is 35ns and cclk is 3.0ns. t_ras * 3ns >= 35ns + 1.5ns The right value for t_ras is 13. The minimum value is 2.	R/W	0x6

1.14.14 AC Timing Register n for the Data of Memory (TimingData n, R/W, Address Offset=0x0038(for n = 0), 0x00E8(for n = 1))

Field	Bit	Description	R/W	Initial State
t_wtr	[31:28]	Internal write to Read command delay, in cclk cycles t_wtr * T(cclk) should be greater than or equal to the minimum value of memory tWTR In LPDDR2-S4/LPDDR3 t_wtr is max(2tCK, tWTR) andIn DDR3 t_wtr is max(4tCK, tWTR). And then this value should be changed in cclk cycles. The minimum value is 2.	R/W	0x1
t_wr	[27:24]	Write recovery time, in cclk cycles t_wr * T(cclk) should be greater than or equal to the minimum value of memory tWR. The minimum value is 2.	R/W	0x2
t_rtp	[23:20]	Internal read to Precharge command delay, in cclk cycles t_rtp * T(cclk) should be greater than or equal to the minimum value of memory tRTP. The minimum value is 2.	R/W	0x3
Reserved	[19:18]	Should be zero		0x0
t_ppd	[17]	Precharge to Precharge Delay 0x0 = tPPD is 1 (for LPDDR3e 1600) 0x1 = tPPD is 2 (for LPDDR3e 1866/2133)	R/W	0x0
Reserved	[16:15]	Should be zero		0x0
t_w2w_c2c	[14]	Additional Write to Write delay in chip to chip case in cclk cycles. The default value of zero means that DREX puts no idle cclk cycle between write command to one chip and write command to other chip. Increase t_w2w_c2c value to put more idle cclk cycle. If mem_term_en of PhyControl0 register(offset addr = 0x18) is enabled, then set this field to 1 or more. Note that this parameter should have the same value in timing parameter register 0 and 1.	R/W	0x0
Reserved	[13]	Should be zero		0x0
t_r2r_c2c	[12]	Additional Read to Read delay in chip to chip case in cclk cycles. The default value of zero means that DREX puts 1 idle cclk cycle between read command to one chip and read command to other chip. Increase t_r2r_c2c value to put more idle cclk cycle. If mem_term_chips of PhyControl0 register(offset addr = 0x18) is enabled, then set this field to 1 or more. In case of LPDDR3, if ctrl_read and ctrl_gate is controlled by DREX, then set this field to 1. Note that this parameter should have the same value in timing parameter register 0 and 1.	R/W	0x0

wl	[11:8]	Write data latency in memory clock cycles wl should be greater than or equal to the minimum value of memory WL. There is no restriction with LPDDR2-S4 but there is restriction with LPDDR3 and DDR3 like below. In LPDDR3, the minimum wl is 5 and in DDR3, the minimum wl is 6. Note that this parameter should have the same value in timing parameter register 0 and 1.	R/W	0x6
dqsck	[7:4]	tDQSCK in memory clock cycles In DDR3, this value should be set to 0. In LPDDR2/3 and Wide IO Memory, this value should set to RU(tDQSCK max/tCK). tDQSCK max is 5.5ns in LPDDR2/3 and 5ns in Wide IO Memory. In LPDDR3-800Mhz(RL 12, WL6), if mem_term_en or phy_term_en is enabled, dqsck in TimingData0/1 register need be set to bigger value than RU(tDQSCK max/tCK).		0x5
rl	[3:0]	Read data latency in memory clock cycles rl should be greater than or equal to the minimum value of memory RL. In Wide IO memory, rl should be set to 3. Note that this parameter should have the same value in timing parameter register 0 and 1.	R/W	0xC

NOTE: tDAL (Auto precharge write recovery + precharge time) = t_wr + t_rp (automatically calculated)

1.14.15 AC Timing Register n for the Power modes of Memory (TimingPower n, R/W, Address Offset=0x003C(for n = 0), 0x00EC(for n = 1))

Field	Bit	Description	R/W	Initial State
t_faw	[31:26]	Four Active Window(for LPDDR2-S4/LPDDR3/DDR3) Two Active Window(t_taw, for Wide IO Memory) t_faw(t_taw) * T(cclk) should be greater than or equal to the minimum value of memory tFAW(tTAW for Wide IO Memory). The minimum value is 2.	R/W	0xE
t_xsr	[25:16]	Self refresh exit power down to next valid command delay, in cycles t_xsr * T(cclk) should be greater than or equal to the minimum value of memory tXSR. In DDR3, this value should be set to tXSDLL. The minimum value is 2.	R/W	0x1B
t_xp	[15:8]	Exit power down to next valid command delay, in cycles t_xp * T(cclk) should be greater than or equal to the minimum value of memory tXP In DDR3, this value should set to tXPDLL. In DDR3 even though “fast exit” is programmed in MRS, tXPDLL is applied. In DDR3, tXPDLL is likely max(10nCK,24ns). So note that t_xp should set to max(5nCCLK,24ns/CCLK period). The minimum value is 2.	R/W	0x4
t_cke	[7:4]	CKE minimum pulse width (minimum power down mode duration), in cycles t_cke should be greater than or equal to the minimum value of memory tCKE. The minimum value is 2.	R/W	0x2
t_mrd	[3:0]	Mode Register Set command period, in cycles t_mrd should be greater than or equal to the minimum value of memory tMRD In DDR3, this parameter should be set to tMOD value. The minimum value is 2.	R/W	0x2

1.14.16 PHY Status Register (PhyStatus, Read Only, Address Offset=0x0040)

Field	Bit	Description	R/W	Initial State
Reserved	[31:15]	Should be zero		0x0
rdlvl_complete	[14]	Read Level Completion This field is only for PHY V5.	R	0x0
Reserved	[13:4]	Should be zero		0x0
dfi_init_complete	[3]	DFI PHY initialization complete 0 = Initialization has not been finished 1 = Initialization has been finished		0x0
Reserved	[2:0]	Should be zero		0x0

1.14.17 ETCTiming Register (ETCTIMING, R/W, Address Offset=0x0044)

Field	Bit	Description	R/W	Initial State
Reserved	[31:14]	Should be zero		0x0
t_mrr	[13:12]	Mode Register Read Command Period in LPDDR2-S4/LPDDR3 The minimum value is 2.	R/W	0x2
Reserved	[11:10]	Should be zero		0x0
t_srr	[9:8]	Status Register Read command period in cclk cycles (Wide IO Memory)	R/W	0x2
t_src	[7:4]	Read of SRR to next valid command in cclk cycles (Wide IO Memory)	R/W	0x7
Reserved	[3:0]	Should be zero		0x0

1.14.18 Memory ChipStatus Register (ChipStatus, Read Only, Address Offset=0x0048)

Field	Bit	Description	R/W	Initial State
Reserved	[31:12]	Should be zero		0x0
chip_sref_state	[11:8]	Chip is in the self-refresh state	R	0x0
chip_pd_state	[7:4]	Chip is in the powerdown state	R	0x0
chip_busy_state	[3:0]	Chip is in the busy state [0] = chip0 busy state [1] = chip1 busy state [2] = chip2 busy state [3] = chip3 busy state	R	0x0

1.14.19 Memory Mode Registers Status Register (MrStatus, Read Only, Address Offset=0x0054)

Field	Bit	Description	R/W	Initial State
Reserved	[31:8]	Should be zero		0x0
mr_status	[7:0]	Mode Registers Status	R	0x0

1.14.20 Quality of Service Control Register n (QosControl n, R/W, Address Offset=0x0060 + 8n (n=0~15, integer))

Field	Bit	Description	R/W	Initial State
Reserved	[31:28]	Should be zero		0x0
cfg_qos_th	[27:16]	QoS Threshold Cycles 0xn = n aclk cycles The matched request(with AxQoS) uses this value for its timeout threshold. If the decreased timeout value is equal to this field, then the priority of the request is increased to emergent page miss level. If this value is set to zero(default value) then emergent page miss is not applied.	R/W	0x000
Reserved	[15:12]	Should be zero		0x0
cfg_qos	[11:0]	QoS Cycles 0xn = n aclk cycles The matched request(with AxQos) uses this value for its timeout counters instead of ConControl.timeout_cnt.	R/W	0xFFF

1.14.21 Timing Set Switch Configuration Register(TimingSetSw, R/W Address Offset=0x00E0)

Field	Bit	Description	R/W	Initial State
Reserved	[31:5]	Should be zero		0x0
timing_set_sw	[4]	**Timing Parameter Set Switch** 0x0 = Use timing parameter set #0 0x1 = Use timing parameter set #1 This field only valid when timing_set_sw_con is 1 which means timing set usage is controlled by SFR. Timing parameter set #0 offset address are 0x34, 0x38 and 0x3C. Timing parameter set #1 offset address are 0xE4, 0xE8, 0xEC.	R/W	0x0
Reserved	[3:1]	Should be zero		0x0
timing_set_sw_con	[0]	**Decision for Timing Parameter Set Switch Control** 0x0 = Switching controlled by external port (port name is timing_set_sw) 0x1 = Switching controlled by SFR (TimingSetSw.timing_set_sw) This field decide that timing parameter set switch control is done by external port or SFR	R/W	0x0

1.14.22 Write Training Configuration Register (WrTraConfig, R/W, Address Offset=0x00F4)

Field	Bit	Description	R/W	Initial State
row_addr	[31:16]	Row Address for Write Training	R/W	0x0
Reserved	[15:5]	Should be zero		0x0
chip	[4]	Chip for Write Training	R/W	0x0
bank	[3:1]	Bank Address for Write Training	R/W	0x0
write_training_en	[0]	Write Training Enable Use this field to issue ACT command. Before setting this field, below things should be finished. Please see PHY manual. - Set write latency before write training(PHY's control register 26) - Set write training mode(PHY's control register 2) 0x0 = Disable, 0x1 = Enable (Issue ACT command) Note that if write training has been finished, issue PALL command using direct command to close the open row before enabling aref_en in ConControl register.	R/W	0x0

1.14.23 Read Leveling Configuration Register (RdlvlConfig, R/W, Address Offset=0x00F8)

This register is only for PHY V5.

Field	Bit	Description	R/W	Initial State
Reserved	[31:2]	Should be zero		0x0
ctrl_rdlvl_data_en	[1]	Data eye training enable	R/W	0x0
ctrl_rdlvl_gate_en	[0]	Gate training enable This is only valid for DDR3 case. If LPDDR2-S4/LPDDR3 is used, this field must be set to 0x0. When ctrl_rdlvl_en = 1, Read leveling offset vlaues will be used instead of ctrl_offsetr*. If read leveling is used, this vlaue should be high during operation. This field should be set after dfi_init_complete is asserted. 0x0 = Disable, 0x1 = Enable	R/W	0x0

1.14.24 BRB Reservation Control Register (BRBRSVCONTROL, R/W, Address Offset=0x0100)

Field	Bit	Description	R/W	Initial State
Reserved	[31:8]	Should be zero		0x0
brb_rsv_en_w3	[7]	Enable Write-BRB reservation for AXI port 3	R/W	0x0
brb_rsv_en_w2	[6]	Enable Write-BRB reservation for AXI port 2	R/W	0x0
brb_rsv_en_w1	[5]	Enable Write-BRB reservation for AXI port 1	R/W	0x0
brb_rsv_en_w0	[4]	Enable Write-BRB reservation for AXI port 0	R/W	0x0
brb_rsv_en_r3	[3]	Enable Read-BRB reservation for AXI port 3	R/W	0x0
brb_rsv_en_r2	[2]	Enable Read-BRB reservation for AXI port 2	R/W	0x0
brb_rsv_en_r1	[1]	Enable Read-BRB reservation for AXI port 1	R/W	0x0
brb_rsv_en_r0	[0]	Enable Read-BRB reservation for AXI port 0	R/W	0x0

1.14.25 BRB Reservation Configuation Register (BRBRSVCONFIG, R/W, Address Offset=0x0104)

Field	Bit	Description	R/W	Initial State
brb_rsv_th_w3	[31:28]	Write-BRB reservation threshold for AXI port 3 Write request from AXI port3 does not serviced when Write-BRB reservation is enabled for AXI port 3 and the occupancy of corresponding Write-BRB exceeds brb_rsv_th_w3.	R/W	0x8
brb_rsv_th_w2	[27:24]	Enable Write-BRB reservation for AXI port 2 Write request from AXI port2 does not serviced when Write-BRB reservation is enabled for AXI port 2 and the occupancy of corresponding Write-BRB exceeds brb_rsv_th_w2.	R/W	0x8
brb_rsv_th_w1	[23:20]	Enable Write-BRB reservation for AXI port 1 Write request from AXI port1 does not serviced when Write-BRB reservation is enabled for AXI port 1 and the occupancy of corresponding Write-BRB exceeds brb_rsv_th_w1.	R/W	0x8
brb_rsv_th_w0	[19:16]	Enable Write-BRB reservation for AXI port 0 Write request from AXI port0 does not serviced when Write-BRB reservation is enabled for AXI port 0 and the occupancy of corresponding Write-BRB exceeds brb_rsv_th_w0.	R/W	0x8
brb_rsv_th_r3	[15:12]	Enable Read-BRB reservation for AXI port 3 Read request from AXI port3 does not serviced when Read-BRB reservation is enabled for AXI port 3 and the occupancy of corresponding Read-BRB exceeds brb_rsv_th_w3.	R/W	0x8
brb_rsv_th_r2	[11:8]	Enable Read-BRB reservation for AXI port 2 Read request from AXI port2 does not serviced when Read-BRB reservation is enabled for AXI port 2 and the occupancy of corresponding Read-BRB exceeds brb_rsv_th_w2.	R/W	0x8
brb_rsv_th_r1	[7:4]	Enable Read-BRB reservation for AXI port 1 Read request from AXI port1 does not serviced when Read-BRB reservation is enabled for AXI port 1 and the occupancy of corresponding Read-BRB exceeds brb_rsv_th_w1.	R/W	0x8
brb_rsv_th_r0	[3:0]	Enable Read-BRB reservation for AXI port 0 Read request from AXI port0 does not serviced when Read-BRB reservation is enabled for AXI port 0 and the occupancy of corresponding Read-BRB exceeds brb_rsv_th_w0.	R/W	0x8

1.14.26 BRB QoS Configuation Register (BRBQOSCONFIG, R/W, Address Offset=0x0108)

Field	Bit	Description	R/W	Initial State
Reserved	[31:12]	Should be zero		0x0
brb_qos_timer_dec	[11:0]	BRB timer decrementing size for QoS. The timer for request in BRB decreases by brb_qos_timer_dec for the following cases. - When the BRB is full - When the request is from the AXI port whose data buffer is full.	R/W	0x10

1.14.28 Write Leveling Configuration Register0 (WRLVLCONFIG0, R/W, Address Offset=0x0120)

Field	Bit	Description	R/W	Initial State
Reserved	[31:8]	Should be zero		0x0
t_wlo	[7:4]	**Write Leveling Ouput Delay** t_wlo * T(PCLK) should be greater than or equal to the minimum value of memory tWLO and the minimum value is 1.	R/W	0x1
Reserved	[3:1]	Should be zero		0x0
odt_on	[0]	**Turn On ODT for Write Leveling** 0x0 = ODT Turn off 0x1 = ODT Turn on, This field is only for write leveling. Turn on before write leveling and turn off after write leveling is finished. Write leveling procedure is MRS for Write leveling - ODT on - wrlvl_wrdata_en - Read ctrl_io_rdata - Change SDLL code of PHY. Refer to PHY manual for details.	R/W	0x0

1.14.29 Write Leveling Configuration Register1 (WRLVLCONFIG1, R/W, Address Offset=0x0124)

Field	Bit	Description	R/W	Initial State
Reserved	[31:1]	Should be zero		0x0
wrlvl_wrdata_en	[0]	**Generate dfi_wrdata_en_p0 for Write Leveling** Generate 1cycle pulse of dfi_wrdata_en_p0 for write leveling. Write leveling is supported in DDR3 and LPDDR3. Note that if S/W write this field to 1 then, 1 cycle pulse of dfi_wrdata_en_p0 would be generated. Refer to PHY manual for write leveling.	R/W	0x0

1.14.30 Write Leveling Status Register (WRLVLSTATUS, R/W, Address Offset=0x0128)

Field	Bit	Description	R/W	Initial State
Reserved	[31:5]	Should be zero		0x0
wrlvl_fsm	[4:0]	**Write Leveling Status** 5'b0_0001: FSM IDLE 5'b0_0010: FSM_SETUP 5'b0_0100: FSM_ACCESS 5'b0_1000: FSM_DONE 5'b1_0000: FSM_TWLO wrlvl_eq is valid only when wrlvl_fsm is FSM_IDLE. Refer to PHY manual for write leveling.	R	0x1

1.14.31 PPC Clock Control Register (PPCCLKCON, R/W, Address Offset=0x0130)

Field	Bit	Description	R/W	Initial State
Reserved	[31:1]	Should be zero		0x0
pe-rev_clk_en	[0]	**Performance Event Clock Enable** 0x0 = Disable clock for performance event 0x1 = Enable clock for performance event Note that do not access PPC address (from 0xE000 to 0xE140) before this field is set to 1.	R/W	0x0

		Performance events list numbers: {0x3,0x2,0x1,0x0} = {reserved, read, reserved, write} for port0 FIFO arbitrated {0x7,0x6,0x5,0x4} = {reserved, read, reserved, write} for port1 FIFO arbitrated {0xb,0xa,0x9,0x8} = {reserved, read, reserved, write} for port2 FIFO arbitrated {0xf,0xe,0xd,0xc} = {reserved, read, reserved, write} for port3 FIFO arbitrated {0x13,0x12,0x11,0x10} = read to bank[3:0] for port0 {0x17,0x16,0x15,0x14} = read to bank[7:4] for port0 {0x1b,0x1a,0x19,0x18} = write to bank[3:0] for port0 {0x1f,0x1e,0x1d,0x1c} = write to bank[7:4] for port0 {0x23,0x22,0x21,0x20} = read to bank[3:0] for port1 {0x27,0x26,0x25,0x24} = read to bank[7:4] for port1 {0x2b,0x2a,0x29,0x28} = write to bank[3:0] for port1 {0x2f,0x2e,0x2d,0x2c} = write to bank[7:4] for port1 {0x33,0x32,0x31,0x30} = read to bank[3:0] for port2 {0x37,0x36,0x35,0x34} = read to bank[7:4] for port2 {0x3b,0x3a,0x39,0x38} = write to bank[3:0] for port2 {0x3f,0x3e,0x3d,0x3c} = write to bank[7:4] for port2 {0x43,0x42,0x41,0x40} = read to bank[3:0] for port3 {0x47,0x46,0x45,0x44} = read to bank[7:4] for port3 {0x4b,0x4a,0x49,0x48} = write to bank[3:0] for port3 {0x4f,0x4e,0x4d,0x4c} = write to bank[7:4] for port3 {0x67 .. 0x50} = {reserved .. reserved} {0x6b,0x6a,0x69,0x68} = {reserved, cas scheduled, reserved, page hit} {0x6f,0x6e,0x6d,0x6c} = {reserved, reserved, read transfer, write transfer} {0x71,0x70} = {page hit read, page hit write}, {0x73,0x72} = {reserved, reserved}, {0x77,0x76,0x75,0x74} = {reserved, reserved, chip1, chip0} read or write on port0,		

		{0x7b,0x7a,0x79,0x78} = {reserved, reserved, chip1, chip0} read or write on port1, {0x7f,0x7e,0x7d,0x7c} = {reserved, reserved, chip1, chip0} read or write on port2, {0x83,0x82,0x81,0x80} = {reserved, reserved, chip1, chip0} read or write on port3, {0x84} = {phy initiate update request} {0x85} = reserved} {0x87,0x86} = {PRE on phase1, PRE on phase0} {0x8b,0x8a,0x89,0x88} = {chip1 SREF, chip0 SREF, chip1 PD, chip0 PD} {0x8d,0x8c} = {ACT on phase1, ACT on phase0}, {0x8f ,0x8e} = {reserved, reserved} {0x93,0x92,0x91,0x90} = {axi read request on port3, axi read request on port2, axi read request on port1, axi read request on port0} {0x97,0x96,0x95,0x94} = {axi write request on port3, axi write request on port2, axi write request on port1, axi write request on port0}		

1.14.32 Performance Event Configuration0 Register (PerevConfig0, R/W, Address Offset=0x0134)

Field	Bit	Description	R/W	Initial State
Reserved	[31:8]	Should be zero		0x0
perev0_sel	[7:0]	Performance event0 selection See above performance event number	R/W	0x0

1.14.33 Performance Event Configuration1 Register (PerevConfig1, R/W, Address Offset=0x0138)

Field	Bit	Description	R/W	Initial State
Reserved	[31:8]	Should be zero		0x0
perev1_sel	[7:0]	Performance event1 selection See above performance event number	R/W	0x0

1.14.34 Performance Event Configuration2 Register (Perev2Config, R/W, Address Offset=0x013C)

Field	Bit	Description	R/W	Initial State
Reserved	[31:8]	Should be zero		0x0
perev2_sel	[7:0]	Performance event2 selection See above performance event number	R/W	0x0

1.14.35 Performance Event3 Configuration Register (PerevConfig3, R/W, Address Offset=0x0140)

Field	Bit	Description	R/W	Initial State
Reserved	[31:8]	Should be zero		0x0
perev3_sel	[7:0]	Performance event3 selection See above performance event number	R/W	0x0

1.14.36 CTRL_IO_RDATA Register (CTRL_IO_RDATA, R, Address Offset=0x0150)

Field	Bit	Description	R/W	Initial State
ctrl_io_rdata	[31:0]	ctrl_io_rdata from PHY	R	0x0

1.14.37 CA Calibration Configuration Register0 (CACAL_CONFIG0, R/W, Address Offset=0x0160)

Field	Bit	Description	R/W	Initial State
dfi_address_p0	[31:12]	**dfi_address_p0 value for CA Calibration** This value would be the expected value for comparing the address pattern received from memory.		0x3FF
Reserved	[11:8]	Should be zero		0x0
t_adr	[7:4]	**CSN Low to Data Output Delay** t_adr * T(PCLK) should be greater than or equal to the minimum value of memory tADR and the minimum value is 1.	R/W	0x1
Reserved	[3:1]	Should be zero		0x0
deassert_cke	[0]	**Deassert CKE for CA Calibration** 0x0 = Put CKE pin to normal operation 0x1 = Put CKE pin to low This field is only for CA calibration. Deassert CKE before CA calibration and put to normal operation after CA calibration is finished. CA calibration procedure is MRS for CA calibration - Deassert CKE - cacal_csn - Read ctrl_io_rdata - Change SDLL code of PHY. Refer to PHY manual for details.	R/W	0x0

1.14.38 CA Calibration Configuration Register (CACAL_CONFIG1, R/W, Address Offset=0x0164)

Field	Bit	Description	R/W	Initial State
Reserved	[31:1]	Should be zero		0x0
cacal_csn	[0]	**Generate dfi_csn_p0 for CA Calibration** Generate 1cycle pulse of dfi_csn_p0 for CA calibration. CA calibration is supported in LPDDR3. Note that if S/W writes this field to 1 then, 1 cycle pulse of dfi_csn_p0 would be generated. Refer to PHY manual for CA calibration.	R/W	0x0

1.14.39 CA Calibration Status Register (CACAL_STATUS, R, Address Offset=0x0168)

Field	Bit	Description	R/W	Initial State
Reserved	[31:5]	Should be zero		0x0
cacal_fsm	[4:0]	**Write Leveling Status** 5'b0_0001: FSM IDLE 5'b0_0010: FSM_SETUP 5'b0_0100: FSM_ACCESS 5'b0_1000: FSM_DONE 5'b1_0000: FSM_TADR CTRL_IO_RDATA are valid only when wrlvl_fsm is FSM_IDLE. Refer to PHY manual for CA calibration.	R	0x0

1.14.40 Emergent Configuration Register 0 (EMERGENT_CONFIG0, R/W, Address Offset=0x0200)

Field	Bit	Description	R/W	Initial State
emer-gent_r_con fig	[31:1]	**Priority Configuration for Emergent Read Request** 1'b1 for each bit position = use emergent page miss priotiry 1'b0 for each bit position = use timeout priority If the coming request is emergency request, then that request including all corresponding requests from the same master in BRBs have emergent page miss priority or timeout priority according to this field setting.	R/W	0x0
Reserved	[0]	Should be zero		0x0

1.14.41 Emergent Configuration Register 1 (EMERGENT_CONFIG1, R/W, Address Offset=0x0204)

Field	Bit	Description	R/W	Initial State
emer-gent_w_co nfig	[31:1]	**Priority Configuration for Emergent Write Request** 1'b1 for each bit position = use emergent page miss priotiry 1'b0 for each bit position = use timeout priority If the coming request is emergency request, then that request including all corresponding requests from the same master in BRBs have emergent page miss priority or timeout priority according to this field setting.	R/W	0x0
Reserved	[0]	Should be zero		0x0

1.14.42 Back Pressure Control Register For Port n (BP_CONTROLn, R/W, Address Offset=0x0210 + 0x10n (n=0~3))

Field	Bit	Description	R/W	Initial State
Reserved	[31:2]	Should be zero		0x0
bp_under_emergent	[1]	**Back Pressure Control Under Emergent Requests** 0x0 = Disable, 0x1 = Enable If this field is disabled, pure back pressure signal will present to output port. If this field is enabled, back pressure signal will present to output port only when there is emergent request.	R/W	0x1
bp_en	[0]	**Back Pressure Control Enable** 0x0 = Disable, 0x1 = Enable	R/W	0x0

1.14.43 Back Pressure Configuration Register For Read/Port n (BP_CONFIGn_R, R/W, Address Offset=0x0214 + 0x10n(n=0~3))

Field	Bit	Description	R/W	Initial State
Reserved	[31:30]	Should be zero		0x0
bp_off_th_data	[29:24]	**Back Pressure Off Threshold Configuration For Data Buffer** 0xn = Number of empty data buffer (n = 0x0 ~ 0x3F) This field defines the threshold of the number of empty data buffer for back pressure.	R/W	0x0
Reserved	[23:22]	Should be zero		0x0
bp_on_th_data	[21:16]	**Back Pressure On Threshold Configuration For Data Buffer** 0xn = Number of empty data buffer (n = 0x0 ~ 0x3F) This field defines the threshold of the number of empty data buffer for back pressure.	R/W	0x0
Reserved	[15:11]	Should be zero		0x0
bp_off_th_brb	[10:8]	**Back Pressure Off Threshold Configuration For BRB** 0xn = Number of empty BRB (n = 0x0 ~ 0x7) This field defines the threshold of the number of empty BRB for back pressure.	R/W	0x0
Reserved	[7:3]	Should be zero		0x0
bp_on_th_brb	[2:0]	**Back Pressure On Threshold Configuration For BRB** 0xn = Number of empty BRB slots (n = 0x0 ~ 0x7)	R/W	0x0

		This field defines the threshold of the number of empty BRB for back pressure.		

1.14.44 Back Pressure Configuration Register For Write/Port n (BP_CONFIGn_W, R/W, Address Offset=0x0218 + 0x10n(n=0~3))

Field	Bit	Description	R/W	Initial State
Reserved	[31:30]	Should be zero		0x0
bp_off_th_data	[29:24]	**Back Pressure Off Threshold Configuration For Data Buffer** 0xn = Number of empty data buffer (n = 0x0 ~ 0x3F) This field defines the threshold of the number of empty data buffer for back pressure.	R/W	0x0
Reserved	[23:22]	Should be zero		0x0
bp_on_th_data	[21:16]	**Back Pressure On Threshold Configuration For Data Buffer** 0xn = Number of empty data buffer (n = 0x0 ~ 0x3F) This field defines the threshold of the number of empty data buffer for back pressure.	R/W	0x0
Reserved	[15:11]	Should be zero		0x0
bp_off_th_brb	[10:8]	**Back Pressure Off Threshold Configuration For BRB** 0xn = Number of empty BRB (n = 0x0 ~ 0x7) This field defines the threshold of the number of empty BRB for back pressure.	R/W	0x0
Reserved	[7:3]	Should be zero		0x0
bp_on_th_brb	[2:0]	**Back Pressure On Threshold Configuration For BRB** 0xn = Number of empty BRB (n = 0x0 ~ 0x7) This field defines the threshold of the number of empty BRB for back pressure.	R/W	0x0

1.14.45 Window Configuration for Write ODT Register (WinConfig_W_ODT, R/W, Address Offset=0x0300)

This is optional feature and not verified.

Field	Bit	Description	R/W	Initial State
Reserved	[31:18]	Should be zero		0x0
p1_extend	[17:16]	Extend Write ODT of Phase1 by ACLK Cycles In case of LPDDR3-800Mhz(RL=12, WL=6), the recommend value is 0x2.	R/W	0x0
Reserved	[15:14]	Should be zero		0x0
p0_extend	[13:12]	Extend Write ODT of Phase0 by ACLK Cycles In case of LPDDR3-800Mhz(RL=12, WL=6), the recommend value is 0x1.	R/W	0x0
Reserved	[11:10]	Should be zero		0x0
p1_delay	[9:8]	Delay Write ODT of Phase1 by ACLK Cycles In case of LPDDR3-800Mhz(RL=12, WL=6), the recommend value is 0x1.	R/W	0x0
Reserved	[7:6]	Should be zero		0x0
p0_delay	[5:4]	Delay Write ODT of Phase0 by ACLK Cycles In case of LPDDR3-800Mhz(RL=12, WL=6), the recommend value is 0x2.	R/W	0x0
use_prog_window	[0]	Programmed Window Enable 0x0 = Disable, 0x1 = Enable Use this field to enable programmed window for write odt. This is memory side odt.	R/W	0x0

1.14.46 Window Configuration for CTRLREAD Register (WinConfig_CTRLREAD, R/W, Address Offset=0x0308)

This is optional feature and not verified.

Field	Bit	Description	R/W	Initial State
Reserved	[31:18]	Should be zero		0x0
p1_extend	[17:16]	Extend Write ODT of Phase1 by ACLK Cycles In case of LPDDR3-800Mhz(RL=12, WL=6), the recommend value is 0x2.	R/W	0x0
Reserved	[15:14]	Should be zero		0x0
p0_extend	[13:12]	Extend Write ODT of Phase0 by ACLK Cycles In case of LPDDR3-800Mhz(RL=12, WL=6), the recommend value is 0x2.	R/W	0x0
Reserved	[11:10]	Should be zero		0x0
p1_delay	[9:8]	Delay Write ODT of Phase1 by ACLK Cycles	R/W	0x0

		In case of LPDDR3-800Mhz(RL=12, WL=6), the recommend value is 0x1.		
Reserved	[7:6]	Should be zero		0x0
p0_delay	[5:4]	Delay Write ODT of Phase0 by ACLK Cycles In case of LPDDR3-800Mhz(RL=12, WL=6), the recommend value is 0x1.	R/W	0x0
use_prog_window	[0]	Programmed Window Enable 0x0 = Disable, 0x1 = Enable Use this field to enable programmed window for ctrl_read. This is AP side odt.	R/W	0x0

1.14.47 Window Configuration for CTRLGATE Register (WinConfig_CTRLGATE, R/W, Address Offset=0x030C)

This is optional feature and not verified.

Field	Bit	Description	R/W	Initial State
Reserved	[31:18]	Should be zero		0x0
p1_extend	[17:16]	Extend Write ODT of Phase1 by ACLK Cycles In case of LPDDR3-800Mhz(RL=12, WL=6), the recommend value is 0x2.	R/W	0x0
Reserved	[15:14]	Should be zero		0x0
p0_extend	[13:12]	Extend Write ODT of Phase0 by ACLK Cycles In case of LPDDR3-800Mhz(RL=12, WL=6), the recommend value is 0x2.	R/W	0x0
Reserved	[11:10]	Should be zero		0x0
p1_delay	[9:8]	Delay Write ODT of Phase1 by ACLK Cycles In case of LPDDR3-800Mhz(RL=12, WL=6), the recommend value is 0x1.	R/W	0x0
Reserved	[7:6]	Should be zero		0x0
p0_delay	[5:4]	Delay Write ODT of Phase0 by ACLK Cycles In case of LPDDR3-800Mhz(RL=12, WL=6), the recommend value is 0x1.	R/W	0x0
use_prog_window	[0]	Programmed Window Enable 0x0 = Disable, 0x1 = Enable Use this field to enable programmed window for ctrl_gate.	R/W	0x0

1.14.48 Performance Monitor Control Register (PMNC_PPC, R/W, Address Offset=0xE000)

Field	Bit	Description	R/W	Initial State
Reserved	[31:18]	Should be zero		0x0
START MODE	[16]	PPC Start Mode 0x0 = SW(by CPU) 0x1 = HW(by SYSCON)	R/W	0x0
Reserved	[15:4]	Should be zero		0x0
CC DIVIDER	[3]	Cycle count divider 0x0 = counts every processor clock cycle, reset value 0x1 = counts every 64th processor clock cycle	R/W	0x0
CC RESET	[2]	Cycle counter reset 0x0 = no action 0x1 = resets cycle counter, CCNT, to zero	W	0x0
PPC COUNTER RESET	[1]	Performance counter reset 0x0 = no action 0x1 = resets all performance counters to zero	W	0x0
PPC ENABLE	[0]	Enable bit 0x0 = disables all counters including CCNT 0x1 = enables all counters including CCNT When you read it, 1 means it's counting and 0 means it's idle (stop counting). You can write it only when the start mode is set to be 0 (PPMU is started by CPU). At this time, you can write this bit by 1 to start counting and write it by 0 to stop the counting. When the start mode is set to be 1 (PPMU is started by SYSCON), you only can read it and get the status of the PPMU. At this time, PPMU is controlled by external trigger. When external trigger is 1, counting starts, and when external trigger is 0, counting stops.	R/W	0x0

1.14.49 Count Enable Set Register (CNTENS_PPC, R/W, Address Offset=0xE010)

Field	Bit	Description	R/W	Initial State
CCNT	[31]	Enable Cycle Counter Write 0x0: NOP Write 0x1: Enables cycle counter Read 0x0: Cycle counter is disabled. Read 0x1: Cycle counter is enabled.	R/W	0x0
Reserved	[30:4]	Should be zero		0x0
PMCNT3	[3]	Enable Counter 3 Write 0x0: NOP Write 0x1: Enables performance counter3 Read 0x0: Performance counter3 is disabled Read 0x1: Performance counter3 is enabled	R/W	0x0
PMCNT2	[2]	Enable Counter 2 Write 0x0: NOP Write 0x1: Enables performance counter2 Read 0x0: Performance counter2 is disabled Read 0x1: Performance counter2 is enabled	R/W	0x0
PMCNT1	[1]	Enable Counter 1 Write 0x0: NOP Write 0x1: Enables performance counter1 Read 0x0: Performance counter1 is disabled Read 0x1: Performance counter1 is enabled	R/W	0x0
PMCNT0	[0]	Enable Counter 0 Write 0x0: NOP Write 0x1: Enables performance counter0 Read 0x0: Performance counter0 is disabled Read 0x1: Performance counter0 is enabled	R/W	0x0

1.14.50 Count Enable Clear Register (CNTENC_PPC, R/W, Address Offset=0xE020)

[NOTE] This SFR has different meaning for read and write. Therefore SFR read value can NOT be the same with the previous written value. For example, when you write CNTENC_PPC.PMCNT2 as 0x1, the writing makes performance counter2 stops counting. After you finishing the writing and read it, then the value will be 0x0. Since the meaning when you read it is count enable information.

Field	Bit	Description	R/W	Initial State
CCNT	[31]	Disable Cycle Counter Write 0x0: NOP Write 0x1: Disables cycle counter Read 0x0: Cycle counter is disabled Read 0x1: Cycle counter is enabled	R/W	0x0
Reserved	[30:4]	Should be zero		0x0
PMCNT3	[3]	Disable Counter 3 Write 0x0: NOP Write 0x1: Disables performance counter3 Read 0x0: Performance counter3 is disabled Read 0x1: Performance counter3 is enabled	R/W	0x0
PMCNT2	[2]	Disable Counter 2 Write 0x0: NOP Write 0x1: Disables performance counter2 Read 0x0: Performance counter2 is disabled Read 0x1: Performance counter2 is enabled	R/W	0x0
PMCNT1	[1]	Disable Counter 1 Write 0x0: NOP Write 0x1: Disables performance counter1 Read 0x0: Performance counter1 is disabled Read 0x1: Performance counter1 is enabled	R/W	0x0
PMCNT0	[0]	Disable Counter 0 Write 0x0: NOP Write 0x1: Disables performance counter0 Read 0x0: Performance counter0 is disabled Read 0x1: Performance counter0 is enabled	R/W	0x0

1.14.51 Interrupt Enable Set Register (INTENS_PPC, R/W, Address Offset=0xE030)

Field	Bit	Description	R/W	Initial State
CCNT	[31]	CCNT overflow interrupt enable Write 0x0: NOP Write 0x1: Enables interrupt generation of cycle counter Read 0x0: Interrupt generation of cycle counter is disabled Read 0x1: Interrupt generation of cycle counter is enabled	R/W	0x0
Reserved	[30:4]	Should be zero		0x0
PMCNT3	[3]	Counter 3 overflow interrupt enable Write 0x0: NOP Write 0x1: Enables interrupt generation of performance counter3 Read 0x0: Interrupt generation of performance counter3 is disabled Read 0x1: Interrupt generation of performance counter3 is enabled	R/W	0x0
PMCNT2	[2]	Counter 2 overflow interrupt enable Write 0x0: NOP Write 0x1: Enables interrupt generation of performance counter2 Read 0x0: Interrupt generation of performance counter2 is disabled Read 0x1: Interrupt generation of performance counter2 is enabled	R/W	0x0
PMCNT1	[1]	Counter 1 overflow interrupt enable Write 0x0: NOP Write 0x1: Enables interrupt generation of performance counter1 Read 0x0: Interrupt generation of performance counter1 is disabled Read 0x1: Interrupt generation of performance counter1 is enabled	R/W	0x0
PMCNT0	[0]	Counter 0 overflow interrupt enable Write 0x0: NOP Write 0x1: Enables interrupt generation of performance counter0 Read 0x0: Interrupt generation of performance counter0 is disabled Read 0x1: Interrupt generation of performance counter0 is enabled	R/W	0x0

1.14.52 Interrupt Enable Clear Register (INTENC_PPC, R/W, Address Offset=0xE040)

[NOTE] This SFR has different meaning for read and write. Therefore SFR read value can NOT be the same with the previous written value. For example, when you write INTENC_PPC.PMCNT2 as 0x1, the writing makes interrupt for counter2 to be disable. After you finishing the writing and read it, then the value will be 0x0. Since the meaning when you read it is interrupt information.

Field	Bit	Description	R/W	Initial State
CCNT	[31]	CCNT overflow interrupt disable Write 0x0: NOP Write 0x1: Disables interrupt generation of cycle counter Read 0x0: Interrupt generation of cycle counter is disabled Read 0x1: Interrupt generation of cycle counter is enabled	R/W	0x0
Reserved	[30:4]	Should be zero		0x0
PMCNT3	[3]	Counter 3 overflow interrupt disable Write 0x0: NOP Write 0x1: Disables interrupt generation of performance counter3 Read 0x0: Interrupt generation of performance counter3 is disabled Read 0x1: Interrupt generation of performance counter3 is enabled	R/W	0x0
PMCNT2	[2]	Counter 2 overflow interrupt disable Write 0x0: NOP Write 0x1: Disables interrupt generation of performance counter2 Read 0x0: Interrupt generation of performance counter2 is disabled Read 0x1: Interrupt generation of performance counter2 is enabled	R/W	0x0
PMCNT1	[1]	Counter 1 overflow interrupt disable Write 0x0: NOP Write 0x1: Disables interrupt generation of performance counter1 Read 0x0: Interrupt generation of performance counter1 is disabled Read 0x1: Interrupt generation of performance counter1 is enabled	R/W	0x0
PMCNT0	[0]	Counter 0 overflow interrupt disable Write 0x0: NOP Write 0x1: Disables interrupt generation of performance counter0	R/W	0x0

		Read 0x0: Interrupt generation of performance counter0 is disabled Read 0x1: Interrupt generation of performance counter0 is enabled		

1.14.53 Overflow Flag Status Register (FLAG_PPC, R/W, Address Offset=0xE050)

[NOTE] This SFR has different meaning for read and write. Therefore SFR read value can NOT be the same with the previous written value. For example, when you write FLAG_PPC.PMCNT2 as 0x1, the writing makes clearing interrupt of counter2. After you finishing the writing and read it, then the value will be 0x0. Since the meaning when you read it is interrupt information.

Field	Bit	Description	R/W	Initial State
CCNT	[31]	Cycle counter overflow flag Write 0x0: NOP Write 0x1: Clears the interrupt of cycle counter Read 0x0: Cycle counter dose not generate the interrupt Read 0x1: Cycle counter generates the interrupt	R/W	0x0
Reserved	[30:4]	Should be zero		0x0
PMCNT3	[3]	Counter 3 overflow flag Write 0x0: NOP Write 0x1: Clears the interrupt of performance counter3 Read 0x0: Performance counter3 dose not generate the interrupt Read 0x1: Performance counte3 generates the interrupt	R/W	0x0
PMCNT2	[2]	Counter 2 overflow flag Write 0x0: NOP Write 0x1: Clears the interrupt of performance counter2 Read 0x0: Performance counter2 dose not generate the interrupt Read 0x1: Performance counte2 generates the interrupt	R/W	0x0
PMCNT1	[1]	Counter 1 overflow flag Write 0x0: NOP Write 0x1: Clears the interrupt of performance counter1 Read 0x0: Performance counter1 dose not generate the interrupt Read 0x1: Performance counte1 generates the interrupt	R/W	0x0
PMCNT0	[0]	Counter 0 overflow flag Write 0x0: NOP Write 0x1: Clears the interrupt of performance counter0 Read 0x0: Performance counter0 dose not generate the inter-	R/W	0x0

		rupt Read 0x1: Performance counte0 generates the interrupt		

1.14.54 Cycle Count Register (CCNT_PPC, R/W, Address Offset=0xE100)

Field	Bit	Description	R/W	Initial State
CCNT	[31:0]	CCNT Register contain an event count Write : Overloads the counter as this value. Read: Current counting value.	R/W	0x0

1.14.55 Performance Monitor Count0 Register (PMCNT0_PPC, R/W, Address Offset=0xE110)

Field	Bit	Description	R/W	Initial State
PMCNT	[31:0]	PMCNT Register contain an event count Write : Overloads the counter as this value. Read: Current counting value.	R/W	0x0

1.14.56 Performance Monitor Count1 Register (PMCNT1_PPC, R/W, Address Offset=0xE120)

Field	Bit	Description	R/W	Initial State
PMCNT	[31:0]	PMCNT Register contain an event count Write : Overloads the counter as this value. Read: Current counting value.	R/W	0x0

1.14.57 Performance Monitor Count2 Register (PMCNT2_PPC, R/W, Address Offset=0xE130)

Field	Bit	Description	R/W	Initial State
PMCNT	[31:0]	PMCNT Register contain an event count Write : Overloads the counter as this value. Read: Current counting value.	R/W	0x0

1.14.58 Performance Monitor Count3 Register (PMCNT3_PPC, R/W, Address Offset=0xE140)

Field	Bit	Description	R/W	Initial State
PMCNT	[31:0]	PMCNT Register contain an event count Write : Overloads the counter as this value. Read: Current counting value.	R/W	0x0

1.15 TZASC Register Description

1.15.1 TZASC Register Overview

Register	Offset	R/W	Description	Initial Value
TZCONFIG	0x0000	R/O	TZASC Configuration Register	0x0000_2208
TZACTION	0x0004	R/W	TZASC Action Register	0x0000_0000
TZLDRANGE	0x0008	R/W	TZASC Lockdown Range Register	0x0000_0000
TZLDSELECT	0x000C	R/W	TZASC Lockdown Select Register	0x0000_0000
TZINTSTATUS	0x0010	R/O	TZASC Interrupt Status Register	0x0000_0000
TZINTCLEAR	0x0014	W/O	TZASC Interrupt Clear Register	0x0000_0000
TZFAILADDRLOWR0	0x0040	R/O	TZASC Read Fail Address Low Register 0	0x0000_0000
TZFAILADDRHIGHR0	0x0044	R/O	TZASC Read Fail Address High Register 0	0x0000_0000
TZFAILCTRLR0	0x0048	R/O	TZASC Read Fail Control Register 0	0x0000_0000
TZFAILIDR0	0x004C	R/O	TZASC Read Fail ID Register 0	0x0000_0000
TZFAILADDRLOWW0	0x0050	R/O	TZASC Write Fail Address Low Register 0	0x0000_0000
TZFAILADDRHIGHW0	0x0054	R/O	TZASC Write Fail Address High Register 0	0x0000_0000
TZFAILCTRLW0	0x0058	R/O	TZASC Write Fail Control Register 0	0x0000_0000
TZFAILIDW0	0x005C	R/O	TZASC Write Fail ID Register 0	0x0000_0000
TZFAILADDRLOWR1	0x0060	R/O	TZASC Read Fail Address Low Register 1	0x0000_0000
TZFAILADDRHIGHR1	0x0064	R/O	TZASC Read Fail Address High Register 1	0x0000_0000
TZFAILCTRLR1	0x0068	R/O	TZASC Read Fail Control Register 1	0x0000_0000
TZFAILIDR1	0x006C	R/O	TZASC Read Fail ID Register 1	0x0000_0000
TZFAILADDRLOWW1	0x0070	R/O	TZASC Write Fail Address Low Register 1	0x0000_0000
TZFAILADDRHIGHW1	0x0074	R/O	TZASC Write Fail Address High Register 1	0x0000_0000
TZFAILCTRLW1	0x0078	R/O	TZASC Write Fail Control Register 1	0x0000_0000
TZFAILIDW1	0x007C	R/O	TZASC Write Fail ID Register 1	0x0000_0000
TZFAILADDRLOWR2	0x0080	R/O	TZASC Read Fail Address Low Register 2	0x0000_0000
TZFAILADDRHIGHR2	0x0084	R/O	TZASC Read Fail Address High Register 2	0x0000_0000
TZFAILCTRLR2	0x0088	R/O	TZASC Read Fail Control Register 2	0x0000_0000
TZFAILIDR2	0x008C	R/O	TZASC Read Fail ID Register 2	0x0000_0000
TZFAILADDRLOWW2	0x0090	R/O	TZASC Write Fail Address Low Register 2	0x0000_0000
TZFAILADDRHIGHW2	0x0094	R/O	TZASC Write Fail Address High Register 2	0x0000_0000
TZFAILCTRLW2	0x0098	R/O	TZASC Write Fail Control Register 2	0x0000_0000
TZFAILIDW2	0x009C	R/O	TZASC Write Fail ID Register 2	0x0000_0000
TZFAILADDRLOWR3	0x00A0	R/O	TZASC Read Fail Address Low Register 3	0x0000_0000
TZFAILADDRHIGHR3	0x00A4	R/O	TZASC Read Fail Address High Register 3	0x0000_0000

TZFAILCTRLR3	0x00A8	R/O	TZASC Read Fail Control Register 3	0x0000_0000
TZFAILIDR3	0x00AC	R/O	TZASC Read Fail ID Register 3	0x0000_0000
TZFAILADDRLOWW3	0x00B0	R/O	TZASC Write Fail Address Low Register 3	0x0000_0000
TZFAILADDRHIGHW3	0x00B4	R/O	TZASC Write Fail Address High Register 3	0x0000_0000
TZFAILCTRLW3	0x00B8	R/O	TZASC Write Fail Control Register 3	0x0000_0000
TZFAILIDW3	0x00BC	R/O	TZASC Write Fail ID Register 3	0x0000_0000
TZRSLOW0	0x0100	R/W	TZASC Region Setup Low Register 0	0x0000_0000
TZRSHIGH0	0x0104	R/W	TZASC Region Setup High Register 0	0x0000_0000
TZRSATTR0	0x0108	R/W	TZASC Region Setup Attribute Register 0	0xF000_0000
TZRSLOW1	0x0110	R/W	TZASC Region Setup Low Register 1	0x0000_0000
TZRSHIGH1	0x0114	R/W	TZASC Region Setup High Register 1	0x0000_0000
TZRSATTR1	0x0118	R/W	TZASC Region Setup Attribute Register 1	0xF000_0000
TZRSLOW2	0x0120	R/W	TZASC Region Setup Low Register 2	0x0000_0000
TZRSHIGH2	0x0124	R/W	TZASC Region Setup High Register 2	0x0000_0000
TZRSATTR2	0x0128	R/W	TZASC Region Setup Attribute Register 2	0xF000_0000
TZRSLOW3	0x0130	R/W	TZASC Region Setup Low Register 3	0x0000_0000
TZRSHIGH3	0x0134	R/W	TZASC Region Setup High Register 3	0x0000_0000
TZRSATTR3	0x0138	R/W	TZASC Region Setup Attribute Register 3	0xF000_0000
TZRSLOW4	0x0140	R/W	TZASC Region Setup Low Register 4	0x0000_0000
TZRSHIGH4	0x0144	R/W	TZASC Region Setup High Register 4	0x0000_0000
TZRSATTR4	0x0148	R/W	TZASC Region Setup Attribute Register 4	0xF000_0000
TZRSLOW5	0x0150	R/W	TZASC Region Setup Low Register 5	0x0000_0000
TZRSHIGH5	0x0154	R/W	TZASC Region Setup High Register 5	0x0000_0000
TZRSATTR5	0x0158	R/W	TZASC Region Setup Attribute Register 5	0xF000_0000
TZRSLOW6	0x0160	R/W	TZASC Region Setup Low Register 6	0x0000_0000
TZRSHIGH6	0x0164	R/W	TZASC Region Setup High Register 6	0x0000_0000
TZRSATTR6	0x0168	R/W	TZASC Region Setup Attribute Register 6	0xF000_0000
TZRSLOW7	0x0170	R/W	TZASC Region Setup Low Register 7	0x0000_0000
TZRSHIGH7	0x0174	R/W	TZASC Region Setup High Register 7	0x0000_0000
TZRSATTR7	0x0178	R/W	TZASC Region Setup Attribute Register 7	0xF000_0000
TZRSLOW8	0x0180	R/W	TZASC Region Setup Low Register 8	0x0000_0000
TZRSHIGH8	0x0184	R/W	TZASC Region Setup High Register 8	0x0000_0000
TZRSATTR8	0x0188	R/W	TZASC Region Setup Attribute Register 8	0xF000_0000
TZITCRG	0x0E00	R/W	TZASC Integration Test Control Register	0x0000_0000
TZITIP	0x0E04	R/O	TZASC Integration Test Input Register	0x0000_0000
TZITOP	0x0E08	R/W	TZASC Integration Test Output Register	0x0000_0000
MEMBASECONFIG0	0x0F00	R/W	Memory Chip0 Base Configuration Register	0x0020_07F8
MEMBASECONFIG1	0x0F04	R/W	Memory Chip1 Base Configuration Register	0x0028_07F8

MEMCONFIG0	0x0F10	R/W	Memory Chip0 Configuration Register	0x0002_2312
MEMCONFIG1	0x0F14	R/W	Memory Chip1 Configuration Register	0x0002_2312

1.15.2 TZASC Configuration Register (TZCONFIG, R/O, Address Offset=0x0000)

Field	Bit	Description	R/W	Initial State
Reserved	[31:14]	Read as zero.	R/O	0x0
addr_width	[13:8]	Address width: - 0x22: 35bit - Others: reserved	R/O	0x22
no_of_regions	[3:0]	Number of regions - 0x8: 9 regions - Others: reserved	R/O	0x8

1.15.3 TZASC Action Register (TZACTION, R/W, Address Offset=0x0004)

Field	Bit	Description	R/W	Initial State
Reserved	[31:2]	Should be zero.	R/W	0x0
reaction_value	[1:0]	Controls how the TZASC uses the bresps[1:0], rresps[1:0], and tzasc_int signals when a region permission failure occurs: b00 = sets tzasc_int LOW and issues an OKAY response b01 = sets tzasc_int LOW and issues a DECERR response b10 = sets tzasc_int HIGH and issues an OKAY response b11 = sets tzasc_int HIGH and issues a DECERR response.	R/W	0x0

1.15.4 TZASC Lockdown Range Register (TZLDRANGE, R/W, Address Offset=0x0008)

Field	Bit	Description	R/W	Initial State
enable	[31]	When set to 1, it enables the lockdown_regions field to control the regions that are to be locked	R/W	0x0
Reserved	[30:9]	Should be zero.	R/W	0x0
lockdown_region	[8:0]	Select regions to lockdown lockdown_region[k] (k=0,…,8): - 1 : Select region k lockdown - 0 : Deselect region k lockdown	R/W	0x0

1.15.5 TZASC Lockdown Select Register (TZLDSELECT, R/W, Address Offset=0x000C)

Field	Bit	Description	R/W	Initial State
Reserved	[30:1]	Should be zero.	R/W	0x0

lock-down_select_region_register	[0]	Modifies the access type of the TZASC Lockdown Range Register when lockdown is triggered: - 1: TZASC Lockdown Range Register is RO. - 0: no effect. TZASC Lockdown Range Register remains RW.	R/W	0x0

1.15.6 TZASC Interrupt Status Register (TZINTSTATUS, R/O, Address Offset=0x0010)

Field	Bit	Description	R/W	Initial State
Reserved	[30:16]	Should be zero.	R/O	0x0
overrun_w3	[15]	When set to 1, it indicates the occurrence of two or more region permission failures for write transactions of AXI port 3 since the interrupt was last cleared	R/O	0x0
status_w3	[14]	Interrupt status for write transactions of AXI port 3 - 1: interrupt is active - 0: interrupt is inactive	R/O	0x0
overrun_r3	[13]	When set to 1, it indicates the occurrence of two or more region permission failures for read transactions of AXI port 3 since the interrupt was last cleared	R/O	0x0
status_r3	[12]	Interrupt status for read transactions of AXI port 3 - 1: interrupt is active - 0: interrupt is inactive	R/O	0x0
overrun_w2	[11]	When set to 1, it indicates the occurrence of two or more region permission failures for write transactions of AXI port 2 since the interrupt was last cleared	R/O	0x0
status_w2	[10]	Interrupt status for write transactions of AXI port 2 - 1: interrupt is active - 0: interrupt is inactive	R/O	0x0
overrun_r2	[9]	When set to 1, it indicates the occurrence of two or more region permission failures for read transactions of AXI port 2 since the interrupt was last cleared	R/O	0x0
status_r2	[8]	Interrupt status for read transactions of AXI port 2 - 1: interrupt is active - 0: interrupt is inactive	R/O	0x0
overrun_w1	[7]	When set to 1, it indicates the occurrence of two or more region permission failures for write transactions of AXI port 1 since the interrupt was last cleared	R/O	0x0
status_w1	[6]	Interrupt status for write transactions of AXI port 1 - 1: interrupt is active - 0: interrupt is inactive	R/O	0x0
overrun_r1	[5]	When set to 1, it indicates the occurrence of two or more region permission failures for read transactions of AXI port 1 since the interrupt was last cleared	R/O	0x0
status_r1	[4]	Interrupt status for read transactions of AXI port 1 - 1: interrupt is active - 0: interrupt is inactive	R/O	0x0
overrun_w0	[3]	When set to 1, it indicates the occurrence of two or more region permission failures for write transactions of AXI port 0 since the interrupt was last cleared	R/O	0x0
status_w0	[2]	Interrupt status for write transactions of AXI port 0 - 1: interrupt is active	R/O	0x0

		- 0: interrupt is inactive		
overrun_r0	[1]	When set to 1, it indicates the occurrence of two or more region permission failures for read transactions of AXI port 0 since the interrupt was last cleared	R/O	0x0
status_r0	[0]	Interrupt status for read transactions of AXI port 0 - 1: interrupt is active - 0: interrupt is inactive	R/O	0x0

1.15.7 TZASC Interrupt Clear Register (TZINTCLEAR, W/O, Address Offset=0x0014)

Writing any value to the TZASC Interrupt Clear Register clears interrupt and sets the:

- status bits to 0 in the TZASC Interrupt Status Register
- overrun bits to 0 in the TZASC Interrupt Status Register.

1.15.8 TZASC Read Fail Address Low Register n (TZFAILADDRLOWRn, R/O, Address Offset=0x0040 + 0x20n (n=0~3, integer))

Field	Bit	Description	R/O	Initial State
fail_addr_low	[30:0]	Returns the AXI address bits [31:0] of the first read access through port n to fail a region permission check after the interrupt was cleared.	R/O	0x0

1.15.9 TZASC Read Fail Address High Register n (TZFAILADDRHIGHRn, R/O, Address Offset=0x0044 + 0x20n (n=0~3, integer))

Field	Bit	Description	R/O	Initial State
Reserved	[31:3]	Read as zero.	R/O	0x0
fail_addr_high	[2:0]	Returns the AXI address bits [34:32] of the first read access through port n to fail a region permission check after the interrupt was cleared.	R/O	0x0

1.15.10 TZASC Read Fail Control Register n (TZFAILCTRLRn, R/O, Address Offset=0x0048 + 0x20n (n=0~3, integer))

Field	Bit	Description	R/O	Initial State
Reserved	[31:25]	Read as zero.	R/O	0x0
write	[24]	This bit indicates whether the first read access through port n to fail a region permission check was a write or read as: - 0: read access (fixed with this value) - 1: write access.	R/O	0x0
Reserved	[23:22]	Read as zero.	R/O	0x0
nonsecure	[21]	After clearing the interrupt status, this bit indicates whether the first read access through port n to fail a region permission check	R/O	0x0

		was non-secure. Read as: - 0: secure access - 1: non-secure access.		
privileged	[20]	After clearing the interrupt status, this bit indicates whether the first read access through port n to fail a region permission check was privileged. Read as: - 0: unprivileged access - 1: privileged access.	R/O	0x0
Reserved	[19:0]	Read as zero.	R/O	0x0

1.15.11 TZASC Read Fail ID Register n (TZFAILIDRn, R/O, Address Offset=0x004C + 0x20n (n=0~3, integer))

Field	Bit	Description	R/O	Initial State
Reserved	[31:16]	Read as zero.	R/O	0x0
axid	[15:0]	Returns the master AXI ID of the first read access through port n to fail a region permission check after the interrupt was cleared	R/O	0x0

1.15.12 TZASC Write Fail Address Low Register n (TZFAILADDRLOWWn, R/O, Address Offset=0x0050 + 0x20n (n=0~3, integer))

Field	Bit	Description	R/O	Initial State
fail_addr_low	[30:0]	Returns the AXI address bits [31:0] of the first write access through port n to fail a region permission check after the interrupt was cleared.	R/O	0x0

1.15.13 TZASC Write Fail Address High Register n (TZFAILADDRHIGHWn, R/O, Address Offset=0x0054 + 0x20n (n=0~3, integer))

Field	Bit	Description	R/O	Initial State
Reserved	[31:3]	Read as zero.	R/O	0x0
fail_addr_high	[2:0]	Returns the AXI address bits [34:32] of the first write access through port n to fail a region permission check after the interrupt was cleared.	R/O	0x0

1.15.14 TZASC Write Fail Control Register n (TZFAILCTRLWn, R/O, Address Offset=0x0058 + 0x20n (n=0~3, integer))

Field	Bit	Description	R/O	Initial State
Reserved	[31:25]	Read as zero.	R/O	0x0
write	[24]	This bit indicates whether the first write access through port n to fail a region permission check was a write or read as: - 0: read access - 1: write access. (fixed with this value)	R/O	0x1
Reserved	[23:22]	Read as zero.	R/O	0x0

nonsecure	[21]	After clearing the interrupt status, this bit indicates whether the first write access through port n to fail a region permission check was non-secure. Read as: - 0: secure access - 1: non-secure access.	R/O	0x0
privileged	[20]	After clearing the interrupt status, this bit indicates whether the first write access through port n to fail a region permission check was privileged. Read as: - 0: unprivileged access - 1: privileged access.	R/O	0x0
Reserved	[19:0]	Read as zero.	R/O	0x0

1.15.15 TZASC Write Fail ID Register n (TZFAILIDWn, R/O, Address Offset=0x005C + 0x20n (n=0~3, integer))

Field	Bit	Description	R/O	Initial State
Reserved	[31:16]	Read as zero.	R/O	0x0
axid	[15:0]	Returns the master AXI ID of the first write access through port n to fail a region permission check after the interrupt was cleared	R/O	0x0

1.15.16 TZASC Region Setup Low Register n (TZRSLOWn, R/W, Address Offset=0x0100 + 0x10n (n=0~8, integer))

Field	Bit	Description	R/W	Initial State
base_address_low	[30:15]	The base address [31:15] of region n. For region 0, this field is Read Only (RO). The base address of region 0 is fixed as 0x0	R/W	0x0
Reserved	[16:0]	Should be zero.	R/W	0x0

1.15.17 TZASC Region Setup High Register n (TZRSHIGHn, R/W, Address Offset=0x0104 + 0x10n (n=0~8, integer))

Field	Bit	Description	R/W	Initial State
Reserved	[31:3]	Should be zero.	R/W	0x0
base_address_high	[2:0]	The base address [34:32] of region n. For region 0, this field is Read Only (RO). The base address of region 0 is fixed as 0x0	R/W	0x0

1.15.18 TZASC Region Attribute Register n (TZRSATTRn, R/W, Address Offset=0x0108 + 0x10n (n=0~8, integer))

Field	Bit	Description	R/W	Initial State
sp	[31:28]	Permission setting for region n. - sp[3]: 1: permits secure read; 0: not permits secure read. - sp[2]: 1: permits secure write; 0: not permits secure write.	R/W	0xF

		- sp[1]: 1: permits nonsecure read; 0: not permits nonsecure read. - sp[0]: 1: permits nonsecure write; 0: not permits nonsecure write.		
Reserved	[27:23]	Should be zero.	R/W	0x0
size	[22:4]	Controls the size of region n. The region size is size*64KB. The maximum region size is 4GB. For region 0, this field is reserved. The region 0 covers entire address space.	R/W	0x0
Reserved	[3:1]	Should be zero.	R/W	0x0
en	[0]	Enables region n. For region 0, this field is reserved. The region 0 is always enabled.	R/W	0x0

1.15.19 TZASC Integration Test Control Register (TZITCRG, R/W, Address Offset=0x0E00)

Field	Bit	Description	R/W	Initial State
Reserved	[31:1]	Should be zero.	R/W	0x0
int_test_en	[0]	Controls the enabling of, or provides the status of, the integration test logic: - 0: integration test logic is disabled - 1: integration test logic is enabled.	R/W	0x0

1.15.20 TZASC Integration Test Input Register (TZITIP, R/O, Address Offset=0x0E04)

Field	Bit	Description	R/O	Initial State
Reserved	[31:1]	Read as zero.	R/O	0x0
secure_boot_lock	[0]	Returns the status of secure_boot_lock: - 0: secure_boot_lock is LOW - 1: secure_boot_lock is HIGH.	R/O	0x0

1.15.21 TZASC Integration Test Output Register (TZITOP, R/W, Address Offset=0x0E08)

Field	Bit	Description	R/W	Initial State
Reserved	[31:1]	Should be zero.	R/W	0x0
itop_int	[0]	Set or reset the value of tzasc_int port by writing 1 or 0 into itop_int bit. If you read, the written value can be read back. - 0: tzasc_int is LOW - 1: tzasc_int is HIGH.	R/W	0x0

1.15.22 Memory Chip0 Base Configuration Register (MemBaseConfig0, R/W, Address Offset=0x0F00)

When rank interleaved address mapping is disabled (MemConfig0/MemConfig1), this register is used for specifying which address region gets mapped to chip 0. When rank interleaving is enabled, the chip number is

obtained from the lower significant bits for rank interleaving, and this register is only used for checking the incoming address's validity.

Field	Bit	Description	R/W	Initial State
Reserved	[31:27]	Should be zero		0x0
chip_base	[26:16]	**AXI Base Address** AXI base address [34:24] = chip_base, For example, if chip_base = 0x20, then AXI base address of memory chip0 becomes 0x2000_0000. It is not necessary that chip_base is aligned with chip size.	R/W	0x20
Reserved	[15:11]	Should be zero		0x0
chip_mask	[10:0]	**AXI Base Address Mask** Upper address bit mask to determine AXI offset address of memory chip0 in assumption of the chip_base is 0x0000_0000. 0 = Corresponding address bit is not to be used for comparison 1 = Corresponding address bit is to be used for comparison For example, if chip_mask = 0x7F8, then AXI offset address becomes 0x0000_0000 ~ 0x07FF_FFFF. If AXI base address of memory chip0 is 0x2000_0000, then memory chip0 has an address range of 0x2000_0000 ~ 0x27FF_FFFF. Simply put, this register represents the size of the chip. The chip size can be obtained by inverting the chip_mask bits, adding 1, and then shifting the value to the left by 24. For example, – If chip size is 256MB, then chip_mask is 0x7F0. – If chip size is 512MB, then chip_mask is 0x7E0. – If chip size is 1GB, then chip_mask is 0x7C0. – If chip size is 2GB, then chip_mask is 0x780. – If chip size is 4GB, then chip_mask is 0x700.	R/W	0x7F8

1.15.23 Memory Chip1 Base Configuration Register (MemBaseConfig1, R/W, Address Offset=0x0F04)

When rank interleaved address mapping is disabled (MemConfig0/MemConfig1), this register is used for specifying which address region gets mapped to chip 1. When rank interleaving is enabled, the chip number is obtained from the lower significant bits for rank interleaving, and this register is only used for checking the incoming address's validity.

Field	Bit	Description	R/W	Initial State
Reserved	[31:27]	Should be zero		0x0
chip_base	[26:16]	**AXI Base Address** AXI base address [34:24] = chip_base, For example, if chip_base = 0x20, then AXI base address of memory chip0 becomes 0x2000_0000. It is not necessary that chip_base is aligned with chip size.	R/W	0x28
Reserved	[15:11]	Should be zero		0x0
chip_mask	[10:0]	**AXI Base Address Mask** Upper address bit mask to determine AXI offset address of memory chip0 in assumption of the chip_base is 0x0000_0000. 0 = Corresponding address bit is not to be used for comparison 1 = Corresponding address bit is to be used for comparison For example, if chip_mask = 0x7F8, then AXI offset address becomes 0x0000_0000 ~ 0x07FF_FFFF. If AXI base address of memory chip0 is 0x2000_0000, then memory chip0 has an address range of 0x2000_0000 ~ 0x27FF_FFFF. Simply put, this register represents the size of the chip. The chip size can be obtained by inverting the chip_mask bits, adding 1, and then shifting the value to the left by 24. For example, – If chip size is 256MB, then chip_mask is 0x7F0. – If chip size is 512MB, then chip_mask is 0x7E0. – If chip size is 1GB, then chip_mask is 0x7C0. – If chip size is 2GB, then chip_mask is 0x780. – If chip size is 4GB, then chip_mask is 0x700.	R/W	0x7F8

1.15.24 Memory Chip0 Configuration Register (MemConfig0, R/W, Address Offset=0x0F10)

Field	Bit	Description	R/W	Initial State
Reserved	[31:23]	Should be zero		0x0
bank_lsb	[22:20]	**LSB of Bank Bit Position in Complex Interleaved Mapping** 0x0 = bit position [8] (column low size = 256B) 0x1 = bit position [9] (column low size = 512B) 0x2 = bit position [10] (column low size = 1KB) 0x3 = bit position [11] (column low size = 2KB) 0x4 = bit position [12] (column low size = 4KB)	R/W	0x0

		0x5 = bit position [13] (column low size = 8KB) Note that column low size should not be bigger than actual memory page size. If rank_inter_en is enabled, then all bank_lsb field of MemConfig0/1 should have the same value.		
rank_inter_en	[19]	**Rank Interleaved Address Mapping** This bit enables chip interleaved address mapping. Note that if this field is 1, two chips configuration should be the same. All rank_inter_en field of MemConfig0/1 should have the same value.	R/W	0x0
bit_sel_en	[18]	**Enable Bit Selection for Randomized Interleaved Address Mapping** This bit enables randomized interleaved address mapping.	R/W	0x0
bit_sel	[17:16]	**Bit Selection for Randomized Interleaved Address Mapping** This field represents the AXI address bit position which will be XORed with bank bits for randomized interleaved address mapping. In case of Wide IO, 0x0: bit position = [13:12] (if rank_inter_en is enabled, [14:12]) 0x1: bit position = [19:18] (if rank_inter_en is enabled, [20:18]) 0x2: bit position = [23:22] (if rank_inter_en is enabled, [24:22]) 0x3: bit position = [27:26] (if rank_inter_en is enabled, [28:26]) In case of Other memory types, 0x0: bit position = [14:12] (if rank_inter_en is enabled, [15:12]) 0x1: bit position = [20:18] (if rank_inter_en is enabled, [21:18]) 0x2: bit position = [24:22] (if rank_inter_en is enabled, [25:22]) 0x3: bit position = [28:26] (if rank_inter_en is enabled, [29:26]) Do not set bit_sel 0x0 if bank bit is [14:12]. For example, if chip_col is 0x3(col addr width is 10bit) and bank_lsb is 0x4, then do not set this field to 0x0.	R/W	0x2
chip_map	[15:12]	**Address Mapping Method (AXI to Memory)** 0x0 = Reserved, 0x1 = Reserved, 0x2 = Split column interleaved ({rank, row, column high, bank, column low, width}) 0x3 ~ 0xf = Reserved	R/W	0x2
chip_col	[11:8]	**Number of Column Address Bits** 0x0 = 7 bits (Wide IO Memory use only), 0x1 = 8 bits (Wide IO Memory use only), 0x2 = 9 bits, 0x3 = 10 bits, 0x4 = 11 bits, 0x5 ~ 0xf = Reserved	R/W	0x3
chip_row	[7:4]	**Number of Row Address Bits**	R/W	0x1

		0x0 = 12 bits, 0x1 = 13 bits, 0x2 = 14 bits, 0x3 = 15 bits, 0x4 = 16 bits, 0x5 ~ 0xf = Reserved		
chip_bank	[3:0]	**Number of Banks** 0x0 ~ 0x1 = Reserved, 0x2 = 4 banks, 0x3 = 8 banks, 0x4 ~ 0xf = Reserved	R/W	0x2

1.15.25 Memory Chip1 Configuration Register (MemConfig1, R/W, Address Offset=0x0F14)

Field	Bit	Description	R/W	Initial State
Reserved	[31:19]	Should be zero		0x0
bank_lsb	[22:20]	**LSB of Bank Bit Position in Complex Interleaved Mapping** 0x0 = bit position [8] (column low size = 256B) 0x1 = bit position [9] (column low size = 512B) 0x2 = bit position [10] (column low size = 1KB) 0x3 = bit position [11] (column low size = 2KB) 0x4 = bit position [12] (column low size = 4KB) 0x5 = bit position [13] (column low size = 8KB) Note that column low size should not be bigger than actual memory page size. If rank_inter_en is enabled, then all bank_lsb field of MemConfig0/1 should have the same value.	R/W	0x0
rank_inter_en	[19]	**Rank Interleaved Address Mapping** This bit enables chip interleaved address mapping. Note that if this field is 1, two chips configuration should be the same. All rank_inter_en field of MemConfig0/1 should have the same value.	R/W	0x0
bit_sel_en	[18]	**Enable Bit Selection for Randomized Interleaved Address Mapping** This bit enables randomized interleaved address mapping.	R/W	0x0
bit_sel	[17:16]	**Bit Selection for Randomized Interleaved Address Mapping** This field represents the AXI address bit position which will be XORed with bank bits for randomized interleaved address mapping. In case of Wide IO, 0x0: bit position = [13:12] (if rank_inter_en is enabled, [14:12]) 0x1: bit position = [19:18] (if rank_inter_en is enabled, [20:18]) 0x2: bit position = [23:22] (if rank_inter_en is enabled, [24:22]) 0x3: bit position = [27:26] (if rank_inter_en is enabled, [28:26]) In case of Other memory types, 0x0: bit position = [14:12] (if rank_inter_en is enabled, [15:12]) 0x1: bit position = [20:18] (if rank_inter_en is enabled, [21:18]) 0x2: bit position = [24:22] (if rank_inter_en is enabled, [25:22]) 0x3: bit position = [28:26] (if rank_inter_en is enabled, [29:26]) Do not set bit_sel 0x0 if bank bit is [14:12]. For example, if chip_col is 0x3(col addr width is 10bit) and bank_lsb is 0x4, then do not set this field to 0x0.	R/W	0x2
chip_map	[15:12]	**Address Mapping Method (AXI to Memory)** 0x0 = Reserved, 0x1 = Reserved, 0x2 = Split column interleaved ({rank, row, column high, bank,	R/W	0x2

		column low, width}) 0x3 ~ 0xf = Reserved		
chip_col	[11:8]	**Number of Column Address Bits** 0x0 = 7 bits (For Wide I/O use only), 0x1 = 8 bits (For Wide I/O use only), 0x2 = reserved, 0x3 = 10 bits, 0x4 = 11 bits, 0x5 ~ 0xf = Reserved	R/W	0x3
chip_row	[7:4]	**Number of Row Address Bits** 0x0 = reserved, 0x1 = 13 bits, 0x2 = 14 bits, 0x3 = 15 bits, 0x4 = 16 bits, 0x5 ~ 0xf = Reserved	R/W	0x1
chip_bank	[3:0]	**Number of Banks** 0x0 ~ 0x1 = Reserved, 0x2 = 4 banks, 0x3 = 8 banks, 0x4 ~ 0xf = Reserved	R/W	0x2