**Non-Confidential Proprietary Notice**

This document is protected by copyright and other related rights and the practice or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of ARM. **No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.**

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether implementations infringe any third party patents.

THIS DOCUMENT IS PROVIDED “AS IS”. ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, ARM makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, third party patents, copyrights, trade secrets, or other rights.

This document may include technical inaccuracies or typographical errors.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This document consists solely of commercial items. You shall be responsible for ensuring that any use, duplication or disclosure of this document complies fully with any relevant export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word “partner” in reference to ARM’s customers is not intended to create or refer to any partnership relationship with any other company. ARM may make changes to this document at any time and without notice.

If any of the provisions contained in these terms conflict with any of the provisions of any signed written agreement covering this document with ARM, then the signed written agreement prevails over and supersedes the conflicting provisions of these terms.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the Agreement shall prevail.

Words and logos marked with ® or ™ are registered trademarks or trademarks of ARM Limited or its affiliates in the EU and/or elsewhere. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners. Please follow ARM’s trademark usage guidelines at [http://www.arm.com/about/trademark-usage-guidelines.php](http://www.arm.com/about/trademark-usage-guidelines.php)

Copyright © [2014, 2015], ARM Limited or its affiliates. All rights reserved.


110 Fulbourn Road, Cambridge, England CB1 9NJ.
Confidentiality status
This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by ARM and the party that ARM delivered this document to.

Unrestricted Access is an ARM internal classification.

Product status
The information in this document is Final, that is for a developed product.

Web address
http://www.arm.com
Contents

Preface

Chapter 1 Introduction

Chapter 2 Functional Description
<table>
<thead>
<tr>
<th>Chapter 3</th>
<th>Programmers Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>About this programmers model</td>
</tr>
<tr>
<td>3.2</td>
<td>Register summary</td>
</tr>
<tr>
<td>3.3</td>
<td>Register descriptions</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Appendix A</th>
<th>Signal Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1</td>
<td>Signals list</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Appendix B</th>
<th>Revisions</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.1</td>
<td>Revisions</td>
</tr>
</tbody>
</table>
Preface

This preface introduces the *ARM® CoreLink™ DMC-520 Dynamic Memory Controller Technical Reference Manual*.

It contains the following:

About this book

This book is for the ARM® CoreLink™ DMC-520 Dynamic Memory Controller.

Product revision status

The rm pn identifier indicates the revision status of the product described in this book, for example, r1p2, where:

rm  Identifies the major revision of the product, for example, r1.
pn  Identifies the minor revision or modification status of the product, for example, p2.

Intended audience

This book is written for experienced engineers who want to integrate the delivered ARM DMC-520 product in a System on Chip (SoC) design.

Using this book

This book is organized into the following chapters:

**Chapter 1 Introduction**
This chapter describes the DMC-520.

**Chapter 2 Functional Description**
This chapter describes how the DMC-520 operates.

**Chapter 3 Programmers Model**
This chapter describes the programmers model of the DMC-520.

**Appendix A Signal Descriptions**
This appendix describes the DMC-520 signals.

**Appendix B Revisions**
This appendix describes the technical changes between released issues of this book.

Glossary

The ARM Glossary is a list of terms used in ARM documentation, together with definitions for those terms. The ARM Glossary does not contain terms that are industry standard unless the ARM meaning differs from the generally accepted meaning.

See the ARM Glossary for more information.

Typographic conventions

*italic* Introduces special terminology, denotes cross-references, and citations.

**bold** Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.

*monospace* Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.

*monospace* Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.

*monospace italic* Denotes arguments to monospace text where the argument is to be replaced by a specific value.

*monospace bold* Denotes language keywords when used outside example code.
Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example:

\[
\text{MRC p15, 0, } <Rd>, <CRn>, <CRm>, <\text{Opcode}_2>\]

**SMALL CAPITALS**

Used in body text for a few terms that have specific technical meanings, that are defined in the *ARM glossary*. For example, *IMPLEMENTATION DEFINED*, *IMPLEMENTATION SPECIFIC*, *UNKNOWN*, and *UNPREDICTABLE*.

**Timing diagrams**

The following figure explains the components used in timing diagrams. Variations, when they occur, have clear labels. You must not assume any timing information that is not explicit in the diagrams.

Shaded bus and signal areas are undefined, so the bus or signal can assume any value within the shaded area at that time. The actual level is unimportant and does not affect normal operation.

![Timing Diagram](image)

**Figure 1  Key to timing diagram conventions**

**Signals**

The signal conventions are:

**Signal level**

The level of an asserted signal depends on whether the signal is active-HIGH or active-LOW. Asserted means:

- HIGH for active-HIGH signals.
- LOW for active-LOW signals.

**Lower-case n**

At the start or end of a signal name denotes an active-LOW signal.

**Additional reading**

This book contains information that is specific to this product. See the following documents for other relevant information.
ARM publications
The following confidential books are only available to licensees:

• ARM® CoreLink™ DMC-520 Dynamic Memory Controller Design Manual (ARM 100001).
• ARM® CoreLink™ DMC-520 Dynamic Memory Controller Integration Manual (ARM 100003).
• ARM® CoreLink™ DMC-520 Dynamic Memory Controller Implementation Guide (ARM 100002).
• ARM® AMBA® 5 CHI Protocol Specification (ARM IHI 0050).
• ARM® Low Power Interface Specification, Q-Channel and P-Channel Interfaces (ARM IHI 0068).
• ARM® AMBA® APB Protocol Specification (ARM IHI 0024).

Other publications


——— Note ————
See the ARM® CoreLink™ DMC-520 Dynamic Memory Controller Release Note for the actual versions of the specifications that ARM used when designing the device.
Feedback

Feedback on this product

If you have any comments or suggestions about this product, contact your supplier and give:
• The product name.
• The product revision or version.
• An explanation with as much information as you can provide. Include symptoms and diagnostic procedures if appropriate.

Feedback on content

If you have comments on content then send an e-mail to errata@arm.com. Give:
• The title ARM® CoreLink™ DMC-520 Dynamic Memory Controller Technical Reference Manual.
• The number ARM 100000_0200_01_en.
• If applicable, the page number(s) to which your comments refer.
• A concise explanation of your comments.

ARM also welcomes general suggestions for additions and improvements.

Note

ARM tests the PDF only in Adobe Acrobat and Acrobat Reader, and cannot guarantee the quality of the represented document when used with any other PDF reader.
Chapter 1
Introduction

This chapter describes the DMC-520.

It contains the following sections:

• 1.1 About the product on page 1-12.
• 1.2 DMC-520 compliance on page 1-13.
• 1.3 Features on page 1-14.
• 1.4 Interfaces on page 1-15.
• 1.5 Configurable options on page 1-16.
• 1.6 Test features on page 1-17.
• 1.7 Product documentation and design flow on page 1-18.
• 1.8 Product revisions on page 1-20.
1.1 About the product

This is a high-level overview of the DMC-520.

The DMC-520 is an ARM AMBA 5 CHI SoC peripheral developed, tested, and licensed by ARM. It is a high-performance, area-optimized memory controller that is compatible with the AMBA 5 CHI protocol. It supports the following memory devices:

- **Double Data Rate 3 (DDR3) SDRAM.**
- **Low-voltage DDR3 SDRAM.**
- **Double Data Rate 4 (DDR4) SDRAM.**

The following figure shows an example system.

![Diagram of an example system](image)

The DMC-520 enables data transfer between the SoC and the SDRAM devices external to the chip. It connects to the on-chip system through a single CHI interface and to a processor through the programmers APB3™ interface to program the DMC-520. It connects to the SDRAM devices through its memory interface block and the DDR PHY Interface (DFI).
1.2 **DMC-520 compliance**

The DMC-520 is compatible with the following protocol specifications and standards:

- AMBA 5 CHI protocol.
- AMBA 3 APB protocol.
- JEDEC DDR4 JESD79-4 standard.
- JEDEC DDR3 JESD79-3 standard.
- JEDEC DDR3L JESD79-3-1 standard.
- JEDEC JESD82-29 standard.
- JEDEC LRDIMM DDR4 Memory Buffer Spec Proposal.
- DDR4 SDRAM Registered DIMM Design Specification.
- DDR4 SDRAM Load Reduced DIMM Design Specification.
- DFI 3.1.
1.3 Features

The DMC-520 supports DDR3 and DDR4 SDRAMs. It also supports error checking, reliability, availability, and serviceability features. In addition, Quality of Service (QoS) features and ARM TrustZone® architecture security extensions are built in throughout the controller.

The system interface provides a CHI interface for connection to a CoreLink Cache Coherent Network (CCN), an APB3 interface for configuration and initialization purposes, and an external performance event interface for connecting to CoreSight™ on-chip debug and trace technology.

The DMC-520 has the following features:

• Profiling signals that enable performance profiling to be performed in the system.
• TrustZone architecture security extensions.
• Buffering to optimize read and write turnaround and to maximize bandwidth.
• A system interface that provides:
  — A CHI interface to connect to a CCN.
  — An APB3 interface for configuration and initialization purposes.
• A Memory Interface (MI) that provides:
  — A DFI interface to a PHY that supports DDR3, DDR3L, and DDR4.
  — Support for DFI 1:1, 1:2, and 1:4 DFI frequency ratio modes.
  — Support for either a 32-bit wide data SDRAM interface or a 64-bit wide data SDRAM interface.
• Low-power operation through programmable SDRAM power modes.
• Reliability, Availability, Serviceability (RAS):
  — Single Error Correcting, Double Error Detecting (SEC-DED) ECC for off-chip DRAM.
  — Symbol-based ECC, to correct memory chip and data-lane failures.
  — SEC-DED ECC for on-chip RAM protection.
  — Hardware Read-Modify-Write (RMW) for systems supporting sparse writes.
  — Link protection for DDR4 link errors.
  — CRC write-data protection for DDR4 devices.
• A programmable mechanism for automated SDRAM scrubbing.
• Error handling.
• Refresh Control Logic for memory banks.
• Power Control Logic. This generates power down requests to the SDRAM, and manages power enables for the PHY logic.
1.4 Interfaces

This section lists the interfaces in the DMC-520.

The DMC-520 has the following external interfaces:

• A system interface to provide read and write access to or from a master. It uses the CHI protocol.
• An APB3 programmers interface to program and control the DMC-520.
• A DFI compatible PHY interface to transfer data to and from the external memory.
• A profile and debug interface.
• A low-power clock control interface that uses the Q-channel protocol. See Q-channel interface on page 2-27.
• An abort interface that is a 4-phase request and acknowledge handshake that you can use to recover from a livelock caused by DRAM or PHY failure.
• User I/O ports.
• A set of interrupts used to detect some operational events or handle errors for example.
1.5 Configurable options

The DMC-520 can be configured to support three different DFI frequency ratios.

The different DFI frequency ratios that the DMC can support are defined in the DFI specification. The DMC supports the following options:

- 1:1 Frequency Ratio Mode where dfi clk = SDRAM CLK.
- 1:2 Frequency Ratio Mode where dfi clk = 1/2 SDRAM CLK.
- 1:4 Frequency Ratio Mode where dfi clk = 1/4 SDRAM CLK.

Note

The DFI Frequency Ratio is the only configurable option for the DMC-520.
1.6 Test features

The DMC-520 provides the following test features:

- Integration test logic for integration testing.
- A debug and profile interface to enable you to monitor transaction events.
1.7 Product documentation and design flow

This section describes the DMC books and how they relate to the design flow.

Documentation

The DMC documentation is as follows:


The Technical Reference Manual (TRM) summarizes the functionality of the DMC, and describes its pins.

Design Manual

The Design Manual (DM) describes the functionality and the effects of functional options on the behavior of the DMC. It is required at all stages of the design flow. The choices made in the design flow can mean that some behavior described in the DM is not relevant. If you are programming the DMC then contact:

• The implementer to determine what integration, if any, was performed before implementing the DMC.
• The integrator to determine the pin configuration of the device that you are using.

The DM is a confidential book that is only available to licensees.

Implementation Guide

The Implementation Guide (IG) describes:

• How to synthesize the Register Transfer Level (RTL).
• How to integrate RAM arrays.
• How to run test patterns.
• The processes to sign off the configured design.

The ARM product deliverables include reference scripts and information about using them to implement your design. Reference methodology flows supplied by ARM are example reference implementations. Contact your EDA vendor for EDA tool support.

The IG is a confidential book that is only available to licensees.

Integration Manual

The Integration Manual (IM) describes how to integrate the DMC into a SoC. It includes a description of the pins that the integrator must tie off to connect the DMC into an SoC design or to other IP.

The IM is a confidential book that is only available to licensees.

Design flow

The DMC is delivered as synthesizable RTL. Before it can be used in a product, it must go through the following processes:

Implementation

The implementer synthesizes the RTL to produce a hard macrocell. This includes integrating RAMs into the design.

Integration

The integrator connects the implemented design into a SoC. This includes connecting it to a memory system.

Programming

This is the last process. The system programmer develops the software required to initialize the DMC, and tests the required application software.

Each process:

• Can be performed by a different party.
• Can include implementation and integration choices that affect the behavior and features of the DMC.
The operation of the final device depends on:

**Configuration inputs**
The integrator configures some features of the DMC by tying inputs to specific values. These configurations affect the start-up behavior before any software configuration is made. They can also limit the options available to the software.

**Software programming**
The programmer configures the DMC by programming particular values into registers. This affects the behavior of the DMC.

--- **Note** ---
This manual refers to implementation-defined features. Reference to a feature that is included means that the appropriate pin configuration options are selected. Reference to an enabled feature means one that has also been configured by software.
1.8 Product revisions

This section describes the differences in functionality between product revisions of the DMC-520.

**r0p0**
- First release.

**r0p1**
- Updated DIMM support.
- Updated address mode.
- Updated the scrub engine operation.
- Added a skip function.
- Added update interrupts and DCI update options.
- Added rank mask capability.
- Added pwakeup.
- Updated DFT signals to latest standard.

**r1p0**
- Added 32-bit memory interface support.

**r2p0**
- Added configurable support for 1:1, 1:2, and 1:4 DFI frequency ratio modes.
Chapter 2
Functional Description

This chapter describes how the DMC-520 operates.

It contains the following sections:
• 2.1 About the functions on page 2-22.
• 2.2 Clocking and resets on page 2-24.
• 2.3 Interfaces on page 2-25.
• 2.4 Constraints and limitations of use on page 2-29.
• 2.5 System address conversion on page 2-30.
2.1 About the functions

This section gives a brief description of all of the functions of the device.

The following figure shows a block diagram of the functions of the DMC-520. The colors show the different categories of functions:

- Blue indicates the blocks that are associated with data flow. The System interface is an example.
- Green indicates the blocks that are associated with programming. The Programming interface is an example.
- Orange indicates the blocks that are associated with the quality and efficiency of the communication to its external memory. The QoS engine is an example.

![Figure 2-1 DMC functional block diagram](image)

**System interface**

The DMC-520 interfaces to the rest of the SoC through this interface. This is a standard CHI interface that connects to a CHI Slave Node Interface (SNF). For any attempted accesses that the system makes outside of the programmed address range of the DMC-520, the system interface responds with a non-data error response. According to how you program the DMC-520, it converts the system access information to the correct rank, bank, column, and row access of the external SDRAM that connects to it. The system interface supports TrustZone features to regulate Secure and Non-secure accesses to both Secure and Non-secure regions of memory.

The DMC monitors queue occupancies and dictates whether system requests of any given QoS is to be accepted. Prefetched and Dynamic P-Credit requests are allocated based on a threshold setting, derived from register settings.

**Note**

There is no support for exclusive access in the DMC because the CHI CCN 50x interconnect series IP supports exclusive access requests in the home node.
Memory channel
Through this interface the DMC-520 conducts its data transactions with the SDRAM and regulates the power consumption of the SDRAM. The DMC-520 uses the ECC information that it receives from the SDRAM to maximize the quality of information that it receives from these devices.

Programming interface
Through this interface a master in the system programs the DMC-520. You can define the Secure and Non-secure regions of external memory and also define how the DMC-520 addresses the external memory from the address that the system provides on its system interface. You can also make direct accesses to the SDRAM, for example to initialize it.

Performance monitoring unit interface
You can use the Performance Monitoring Unit (PMU) interface to monitor the performance and power settings for your specific application. This interface allows you to monitor the inner workings of the device and so enables additional information to be viewed.

QoS engine
The DMC-520 provides controls to enable you to adjust its arbitration scheme for your system to maximize the availability of your external memory devices. It provides buffers to re-order system transaction requests. It uses an advanced scheduling algorithm to ensure that traffic going to one memory bank causes minimal disruption to traffic going to a different memory bank. It also schedules transaction requests according to the availability of the destination memory bank. For system access requests to different available memory banks the DMC-520 arbitrates these requests based on the QoS priority initially then on the temporal priority. These memory access requests all compete for control of the external SDRAM bus and SDRAM bank availability.

RAS
RAS features include support for the following:
• SECDED ECC and symbol-based ECC for external DRAM. The symbol-based ECC performs quad symbol correct and multi-symbol detect.
• SECDED ECC of on-chip SRAM buffers within the DMC-520.
• An automated retry of failed read transactions.
• Write-back of corrected errors.
• To reduce memory errors, the DMC-520 supports:
  — Link error protection for the memory interface.
  — Programmable data scrubbing where the DMC-520 periodically detects and corrects data errors in the memory itself.
2.2 Clocking and resets

The DMC-520 normally operates as one synchronous clock domain between the interconnect and the external DDR interface. However, the programming interface can operate asynchronously to this.

This section shows the clock and reset signals that the DMC-520 requires.

Clocks

There are either two or three clock inputs depending on the DFI frequency ratio configuration:

- **clk.** This is the main DMC clock that runs at SDRAM clock frequency. It must run synchronous to, and at the same frequency, as the CHI interface. If the CHI interconnect is not running at SDRAM clock frequency, then a DSSB bridge must be used. When in a 1:1 DFI frequency ratio mode, this clock doubles as the **dfi clk**.
- **dfi_clk.** This clock port only exists if the DMC is in 1:2 or 1:4 mode. The clock runs the DFI interface and connects to both the DMC and the PHY. It must be edge synchronous to **clk**, and run at twice the **clk** frequency if it is in the 1:2 configuration or four-times the **clk** frequency when in the 1:4 configuration.
- **pclk.** This can run asynchronously to **clk** and **dfi clk**.

Reset

Resets must be applied for a minimum duration of 16 clock cycles for each clock domain.

There are two reset inputs. **RESETn** resets both **clk** and **dfi clk** registers and **PRESETn** resets **pclk** registers. The **pclk** domain must be brought out of reset prior to the **clk** and **dfi clk** domains.

---

**Note**

- To assert any DMC-520 reset signal, you must set it LOW.
- To perform a DMC-520 reset, you must assert both reset signals.

---

Related references

2.3 Interfaces

This section describes the interfaces of the DMC-520, as the following figure shows.

![Interfaces of the DMC-520](image)

This section contains the following subsections:

- 2.3.1 System interface on page 2-25.
- 2.3.2 Programming interface on page 2-25.
- 2.3.3 PHY interface on page 2-25.
- 2.3.4 Profile and debug interface on page 2-26.
- 2.3.5 Low-power clock control interface on page 2-26.
- 2.3.6 Abort interface on page 2-28.

2.3.1 System interface

This section describes the function of the System interface.

The System Interface provides protocol conversion between CHI and internal read/write requests. Because CHI is packet-based and a slave node only supports read and write semantics, this translation is straightforward at a transaction level because no transformation function is performed.

Note

The DMC does not support exclusive access because the CHI CCN 5xx interconnect product family IP supports exclusive access requests in its home node.

2.3.2 Programming interface

This section describes the APB3 interface, used for programming the DMC-520.

The AMBA APB3 slave interface allows software to configure the controller and to initialize the memory devices. The APB3 programming interface also provides a means of performing architectural state transitions in addition to querying certain debug and profile information. The interface is a memory-mapped register interface.

2.3.3 PHY interface

The PHY interface provides command scheduling and arbitration, including the generation of any required SDRAM prepare commands, for example, ACTIVATE and PRECHARGE. This section describes the PHY interface in the DMC-520.
The PHY interface is compatible with the DDR standards for DDR4 and DDR3 (including DDR3L). It provides:
  • Command scheduling and arbitration, including generation of any required SDRAM prepare
    commands, for example, ACTIVATE, or PRECHARGE.
  • Automated AUTOREFRESH command generation.
  • SDRAM interface link protection including automated retries for failed commands to ensure the
    correct ordering of those retried commands to SDRAM.
  • Automated SDRAM and PHY logic power control.
  • Profile and debug information.
  • Support for DFI 1:1, 1:2 and 1:4 frequency ratio modes.

### 2.3.4 Profile and debug interface

This section describes the profile and debug interface in the DMC-520.

The DMC-520 provides programmable features that allow system designers and software developers to
fine-tune performance and power settings for their applications. A number of events can be monitored
and the statistics used to fine-tune the performance of the controller by statically, or dynamically, altering
the programmed state.

The information is made available through output pins that the system integrator must connect to an
external monitoring unit.

The following events are monitored:
  • Channel utilization.
  • Channel and chip power state information.
  • Bank utilization.
  • Bank distribution.
  • Activation rate.
  • Read and write turnaround frequency.
  • Read and write buffer fill status and the frequency of full events.
  • Thresholding asserting back pressure.
  • Arbitration decisions made where QoS is prioritized over efficiency.
  • Read-Modify-Write (RMW) frequency.
  • Timeouts and deadline events.

Each event is implemented as a pair of signals, VALID, and either PAYLOAD or a permanently valid
PAYLOAD signal.

The Profile and Debug event interface can be connected to a generic event counter block, where any
combination of the signals can be logged and tracked, depending on your system requirements.

### 2.3.5 Low-power clock control interface

This section describes the clock requirements for the DMC-520.

The DMC-520 provides a low-power control interface using the Q-channel protocol. This is used to
place the DMC into its low-power state, in which state the clock can be removed. The system can use the
APB interface to put the DMC into its low-power state, and take it out of its low-power state.

SDRAM provides a number of power-saving states, as distinct from those of the DMC-520:

1. Idle-ready.
2. Clock stop.
3. Active power down.
4. Precharge power down.
5. Self-Refresh (SR).

All states prohibit commands apart from Idle-ready. From states 2-6, the energy saving increases, but so
does the exit latency from that state. Some SDRAMs do not support dynamic clock stopping or MPD.
Specific commands, together with the clock-enable CKE signal, are used to control states 2-5. Individual CKE pins are required for each chip that requires separate power control.

The features of the DMC-520 include:

- Separate clock and CKE controls for each chip select, with a set of multiplexer options to support standard DIMM configurations.
- Automated power control of SDRAM power modes based on an enable and timer. See 3.3.8 low_power_control_next on page 3-47.
- Clock stop functionality that differs between memory devices. A programmable register controls this behavior. See 3.3.8 low_power_control_next on page 3-47.
- Auto powerdown with minimal or no latency penalty on wake up.
- Auto self-refresh functionality. The time delay before entry to self-refresh can be timed in refresh periods. When in self-refresh, a chip only comes out of self-refresh in response to system commands.
- Software-controlled low-power entry through the APB programming interface.
- A Q-channel interface for hardware to control entry into the SR states. See Q-channel interface on page 2-27.
- A separate low-power interface to allow clock stopping of the programming interface.

Note

The DMC-520 does not allow multiple methods of low power entry, either software or hardware, that is used at the same time. This is a restriction imposed on the system design.

The PHY logic consumes power in standby mode. If the controller is using SDRAM low-power modes, then it indicates to the PHY that it can power down. The wake-up value that the DMC signals to the PHY with the powerdown request determines the level of power state that the PHY enters. The wake-up value is determined from a programmed value that is associated with each SDRAM power-saving state. These states are:

- Idle.
- Power-down.
- Configuration.
- Self-refresh.
- MPD.

Note

The DMC can also indicate that the PHY must power down in the following ways:

- As a direct command from software, with a software-defined wake-up value.
- As part of a Q-channel sequence, with a tie-off defined wake-up value.

Q-channel interface

The DMC has a Q-channel interface that allows an external power controller to place the DMC into a low-power state.

It is a standard Q-channel interface as defined in the ARM® Low Power Interface Specification, Q-Channel and P-Channel Interfaces using the following 4 signals.

- qactive.
- qreqn.
- qacceptn.
- qdeny.

When the DMC receives a request, it puts the DRAM into self_refresh before asserting qacceptn to accept the request that indicates the clk can be stopped.

DMC denies requests to power down using the Q-Channel when geardown_mode is enabled. In this case low-power mode can still be entered using the APB interface.
There is a separate Q-channel interface for the pclk using the following signals:

- qactive_apb.
- qreqn_apb.
- qacceptn_apb.
- qdeny_apb.

The DMC never denies a request to power down the APB clock although it might be delayed based on APB activity.

**Note**

These two interfaces are interrelated and a change on one can cause qactive on the other to be asserted. If this occurs then the power up request must be responded to straight away to allow the request to be serviced.

See *ARM® Low Power Interface Specification, Q-Channel and P-Channel Interfaces*.

### 2.3.6 Abort interface

The abort interface is a 4-phase request and acknowledge handshake that the DMC can use to recover from a livelock caused by a DRAM failure or a PHY failure. When a failure happens, it causes repeated retries of commands on the memory interface.

The following diagram shows the request, acknowledge handshake.

![Abort Interface Timing Diagram](image)

**Figure 2-3** Abort interface timing diagram

The system can issue an abort at any time that puts the DMC into the ABORT architectural state. Software must then restore the memory state. All current system transactions are retried when instructed by software.
2.4 Constraints and limitations of use

The constraints and limitations of the DMC-520 depend on the SDRAMS used, and the interoperability within the PHYs. This, in turn, depends on the DDR Physical Interface (DFI) parameters.

The SDRAMs supported by the DMC-520 are:

- Double Data Rate 3 (DDR3) SDRAM.
- Low-voltage DDR3 SDRAM.
- Double Data Rate 4 (DDR4) SDRAM.

Note

These devices are described in the JEDEC specifications that are global standards for the microelectronics industry.

The DIMMs supported by the DMC-520 are:

- DDR3 UDIMM.
- DDR3 RDIMM.
- DDR4 UDIMM.
- DDR4 RDIMM.
- DDR4 LRDIMM.
- DDR4 3DS.
2.5 System address conversion

This section describes how the DMC-520 transforms the system address to the SDRAM address.

The following figure shows the functions that the DMC-520 uses to transform the address that it receives from the system to the address it presents to the SDRAM.

The following describes the function of the boxes:

**Address map**
Receives the system address and converts it to a suitable form for the Address decode function.

**TrustZone decode**
Decodes invalid address regions.

**Address decode**
Translates its input address to row, rank, bank, and column addresses.

--- Note ---

A transaction is rejected in the following circumstances:
- The transaction fails the TrustZone permission check.
- If Memory Map translation is enabled, the transaction targets a reserved address region. See *Principles of ARM® Memory Maps White Paper*.
- The transaction targets a physical rank that is masked by the Rank mask function.
- The transaction, or decoded address, falls above the allocated DRAM space.

A rejected transaction has no effect on memory. If a transaction is rejected then write data is ignored and read data is returned as zero. Rejected transactions might be given a non-data error response based on the setting of the enable_err_response* bits in the memory_address_max_31_00* registers. A PMU signal reports any failed transactions. It is up to the system to prevent Non-secure masters from determining information about failed Secure transactions using the PMU signals.
Chapter 3
Programmers Model

This chapter describes the programmers model of the DMC-520.

It contains the following sections:

• 3.1 About this programmers model on page 3-32.
• 3.2 Register summary on page 3-33.
• 3.3 Register descriptions on page 3-45.
3.1 About this programmers model

The following information applies to the dmc520 registers:

- The base address is not fixed, and can be different for any particular system implementation. The offset of each register from the base address is fixed.
- Do not attempt to access reserved or unused address locations. Attempting to access these locations can result in Unpredictable behavior.
- Unless otherwise stated in the accompanying text:
  — Do not modify undefined register bits.
  — Ignore undefined register bits on reads.
  — All register bits are reset to the reset value specified in the 3.2 Register summary on page 3-33
- Access type is described as follows:
  
  **RW**  Read and write.
  
  **RO**  Read only.
  
  **WO**  Write only.
### 3.2 Register summary

The following table shows the registers in offset order from the base memory address.

<table>
<thead>
<tr>
<th>Offset</th>
<th>Name</th>
<th>Type</th>
<th>Reset</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x000</td>
<td>memc_status</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.1 memc_status on page 3-45</td>
</tr>
<tr>
<td>0x004</td>
<td>memc_config</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.2 memc_config on page 3-45</td>
</tr>
<tr>
<td>0x008</td>
<td>memc_cmd</td>
<td>WO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.3 memc_cmd on page 3-45</td>
</tr>
<tr>
<td>0x010</td>
<td>address_control_next</td>
<td>RW</td>
<td>0x00030202</td>
<td>32</td>
<td>3.3.4 address_control_next on page 3-46</td>
</tr>
<tr>
<td>0x014</td>
<td>decode_control_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.5 decode_control_next on page 3-46</td>
</tr>
<tr>
<td>0x018</td>
<td>format_control</td>
<td>RW</td>
<td>0x22000213</td>
<td>32</td>
<td>3.3.6 format_control on page 3-46</td>
</tr>
<tr>
<td>0x01C</td>
<td>address_map_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.7 address_map_next on page 3-46</td>
</tr>
<tr>
<td>0x020</td>
<td>low_power_control_next</td>
<td>RW</td>
<td>0x00000020</td>
<td>32</td>
<td>3.3.8 low_power_control_next on page 3-47</td>
</tr>
<tr>
<td>0x024</td>
<td>turnaround_control_next</td>
<td>RW</td>
<td>0x0F0F0F0F</td>
<td>32</td>
<td>3.3.9 turnaround_control_next on page 3-47</td>
</tr>
<tr>
<td>0x028</td>
<td>hit_turnaround_control_next</td>
<td>RW</td>
<td>0x08909FBF</td>
<td>32</td>
<td>3.3.10 hit_turnaround_control_next on page 3-47</td>
</tr>
<tr>
<td>0x030</td>
<td>qos_class_control_next</td>
<td>RW</td>
<td>0x00000FC8</td>
<td>32</td>
<td>3.3.11 qos_class_control_next on page 3-48</td>
</tr>
<tr>
<td>0x034</td>
<td>escalation_control_next</td>
<td>RW</td>
<td>0x00080F03</td>
<td>32</td>
<td>3.3.12 escalation_control_next on page 3-48</td>
</tr>
<tr>
<td>0x038</td>
<td>qv_control_31_00_next</td>
<td>RW</td>
<td>0x76543210</td>
<td>32</td>
<td>3.3.13 qv_control_31_00_next on page 3-48</td>
</tr>
<tr>
<td>0x03C</td>
<td>qv_control_63_32_next</td>
<td>RW</td>
<td>0xFDCA98</td>
<td>32</td>
<td>3.3.14 qv_control_63_32_next on page 3-48</td>
</tr>
<tr>
<td>0x040</td>
<td>rt_control_31_00_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.15 rt_control_31_00_next on page 3-49</td>
</tr>
<tr>
<td>0x044</td>
<td>rt_control_63_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.16 rt_control_63_32_next on page 3-49</td>
</tr>
<tr>
<td>0x048</td>
<td>timeout_control_next</td>
<td>RW</td>
<td>0x00000001</td>
<td>32</td>
<td>3.3.17 timeout_control_next on page 3-49</td>
</tr>
<tr>
<td>0x04C</td>
<td>credit_control_next</td>
<td>RW</td>
<td>0x00000F03</td>
<td>32</td>
<td>3.3.18 credit_control_next on page 3-50</td>
</tr>
<tr>
<td>0x050</td>
<td>write_priority_control_31_00_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.19 write_priority_control_31_00_next on page 3-50</td>
</tr>
<tr>
<td>0x054</td>
<td>write_priority_control_63_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.20 write_priority_control_63_32_next on page 3-50</td>
</tr>
<tr>
<td>0x060</td>
<td>queue_threshold_control_31_00_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.21 queue_threshold_control_31_00_next on page 3-50</td>
</tr>
<tr>
<td>0x064</td>
<td>queue_threshold_control_63_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.22 queue_threshold_control_63_32_next on page 3-51</td>
</tr>
<tr>
<td>0x078</td>
<td>memory_address_max_31_00_next</td>
<td>RW</td>
<td>0x00000010</td>
<td>32</td>
<td>3.3.23 memory_address_max_31_00_next on page 3-51</td>
</tr>
<tr>
<td>0x07C</td>
<td>memory_address_max_43_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.24 memory_address_max_43_32_next on page 3-51</td>
</tr>
<tr>
<td>0x080</td>
<td>access_address_min0_31_00_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.25 access_address_min0_31_00_next on page 3-52</td>
</tr>
<tr>
<td>0x084</td>
<td>access_address_min0_43_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.26 access_address_min0_43_32_next on page 3-52</td>
</tr>
<tr>
<td>Offset</td>
<td>Name</td>
<td>Type</td>
<td>Reset</td>
<td>Width</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------</td>
<td>------</td>
<td>----------------------</td>
<td>-------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>0x088</td>
<td>access_address_max0_31_00_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.27 access_address_max0_31_00_next on page 3-52</td>
</tr>
<tr>
<td>0x08C</td>
<td>access_address_max0_43_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.28 access_address_max0_43_32_next on page 3-52</td>
</tr>
<tr>
<td>0x090</td>
<td>access_address_min1_31_00_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.29 access_address_min1_31_00_next on page 3-53</td>
</tr>
<tr>
<td>0x094</td>
<td>access_address_min1_43_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.30 access_address_min1_43_32_next on page 3-53</td>
</tr>
<tr>
<td>0x098</td>
<td>access_address_max1_31_00_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.31 access_address_max1_31_00_next on page 3-53</td>
</tr>
<tr>
<td>0x09C</td>
<td>access_address_max1_43_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.32 access_address_max1_43_32_next on page 3-54</td>
</tr>
<tr>
<td>0x0A0</td>
<td>access_address_min2_31_00_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.33 access_address_min2_31_00_next on page 3-54</td>
</tr>
<tr>
<td>0x0A4</td>
<td>access_address_min2_43_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.34 access_address_min2_43_32_next on page 3-54</td>
</tr>
<tr>
<td>0x0A8</td>
<td>access_address_max2_31_00_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.35 access_address_max2_31_00_next on page 3-54</td>
</tr>
<tr>
<td>0x0AC</td>
<td>access_address_max2_43_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.36 access_address_max2_43_32_next on page 3-55</td>
</tr>
<tr>
<td>0x0B0</td>
<td>access_address_min3_31_00_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.37 access_address_min3_31_00_next on page 3-55</td>
</tr>
<tr>
<td>0x0B4</td>
<td>access_address_min3_43_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.38 access_address_min3_43_32_next on page 3-55</td>
</tr>
<tr>
<td>0x0B8</td>
<td>access_address_max3_31_00_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.39 access_address_max3_31_00_next on page 3-56</td>
</tr>
<tr>
<td>0x0BC</td>
<td>access_address_max3_43_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.40 access_address_max3_43_32_next on page 3-56</td>
</tr>
<tr>
<td>0x0C0</td>
<td>access_address_min4_31_00_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.41 access_address_min4_31_00_next on page 3-56</td>
</tr>
<tr>
<td>0x0C4</td>
<td>access_address_min4_43_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.42 access_address_min4_43_32_next on page 3-56</td>
</tr>
<tr>
<td>0x0C8</td>
<td>access_address_max4_31_00_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.43 access_address_max4_31_00_next on page 3-56</td>
</tr>
<tr>
<td>0x0CC</td>
<td>access_address_max4_43_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.44 access_address_max4_43_32_next on page 3-57</td>
</tr>
<tr>
<td>0x0D0</td>
<td>access_address_min5_31_00_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.45 access_address_min5_31_00_next on page 3-57</td>
</tr>
<tr>
<td>0x0D4</td>
<td>access_address_min5_43_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.46 access_address_min5_43_32_next on page 3-58</td>
</tr>
<tr>
<td>0x0D8</td>
<td>access_address_max5_31_00_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.47 access_address_max5_31_00_next on page 3-58</td>
</tr>
<tr>
<td>Offset</td>
<td>Name</td>
<td>Type</td>
<td>Reset</td>
<td>Width</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>----------------------------------------</td>
<td>-------</td>
<td>---------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>0x0DC</td>
<td>access_address_max5_43_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.48 access_address_max5_43_32_next on page 3-58</td>
</tr>
<tr>
<td>0x0E0</td>
<td>access_address_min6_31_00_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.49 access_address_min6_31_00_next on page 3-58</td>
</tr>
<tr>
<td>0x0E4</td>
<td>access_address_min6_43_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.50 access_address_min6_43_32_next on page 3-59</td>
</tr>
<tr>
<td>0x0E8</td>
<td>access_address_max6_31_00_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.51 access_address_max6_31_00_next on page 3-59</td>
</tr>
<tr>
<td>0x0EC</td>
<td>access_address_max6_43_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.52 access_address_max6_43_32_next on page 3-59</td>
</tr>
<tr>
<td>0x0F0</td>
<td>access_address_min7_31_00_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.53 access_address_min7_31_00_next on page 3-60</td>
</tr>
<tr>
<td>0x0F4</td>
<td>access_address_min7_43_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.54 access_address_min7_43_32_next on page 3-60</td>
</tr>
<tr>
<td>0x0F8</td>
<td>access_address_max7_31_00_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.55 access_address_max7_31_00_next on page 3-60</td>
</tr>
<tr>
<td>0x0FC</td>
<td>access_address_max7_43_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.56 access_address_max7_43_32_next on page 3-60</td>
</tr>
<tr>
<td>0x100</td>
<td>channel_status</td>
<td>RO</td>
<td>0x00000003</td>
<td>32</td>
<td>3.3.57 channel_status on page 3-61</td>
</tr>
<tr>
<td>0x108</td>
<td>direct_addr</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.58 direct_addr on page 3-61</td>
</tr>
<tr>
<td>0x10C</td>
<td>direct_cmd</td>
<td>WO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.59 direct_cmd on page 3-61</td>
</tr>
<tr>
<td>0x110</td>
<td>dci_replay_type_next</td>
<td>RW</td>
<td>0x00000002</td>
<td>32</td>
<td>3.3.60 dci_replay_type_next on page 3-62</td>
</tr>
<tr>
<td>0x118</td>
<td>dci_strb</td>
<td>RW</td>
<td>0x0000000F</td>
<td>32</td>
<td>3.3.61 dci_strb on page 3-62</td>
</tr>
<tr>
<td>0x11C</td>
<td>dci_data</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.62 dci_data on page 3-62</td>
</tr>
<tr>
<td>0x120</td>
<td>refresh_control_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.63 refresh_control_next on page 3-62</td>
</tr>
<tr>
<td>0x128</td>
<td>memory_type_next</td>
<td>RW</td>
<td>0x00000101</td>
<td>32</td>
<td>3.3.64 memory_type_next on page 3-63</td>
</tr>
<tr>
<td>0x130</td>
<td>feature_config</td>
<td>RW</td>
<td>0x00000F0</td>
<td>32</td>
<td>3.3.65 feature_config on page 3-63</td>
</tr>
<tr>
<td>0x138</td>
<td>nibble_failed_031_000</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.66 nibble_failed_031_000 on page 3-63</td>
</tr>
<tr>
<td>0x13C</td>
<td>nibble_failed_063_032</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.67 nibble_failed_063_032 on page 3-64</td>
</tr>
<tr>
<td>0x140</td>
<td>nibble_failed_095_064</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.68 nibble_failed_095_064 on page 3-64</td>
</tr>
<tr>
<td>0x144</td>
<td>nibble_failed_127_096</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.69 nibble_failed_127_096 on page 3-64</td>
</tr>
<tr>
<td>0x148</td>
<td>queue_allocate_control_031_000</td>
<td>RW</td>
<td>0xFFFFFFF</td>
<td>32</td>
<td>3.3.70 queue_allocate_control_031_000 on page 3-64</td>
</tr>
<tr>
<td>0x14C</td>
<td>queue_allocate_control_063_032</td>
<td>RW</td>
<td>0xFFFFFFF</td>
<td>32</td>
<td>3.3.71 queue_allocate_control_063_032 on page 3-65</td>
</tr>
<tr>
<td>0x150</td>
<td>queue_allocate_control_095_064</td>
<td>RW</td>
<td>0xFFFFFFF</td>
<td>32</td>
<td>3.3.72 queue_allocate_control_095_064 on page 3-65</td>
</tr>
<tr>
<td>0x154</td>
<td>queue_allocate_control_127_096</td>
<td>RW</td>
<td>0xFFFFFFF</td>
<td>32</td>
<td>3.3.73 queue_allocate_control_127_096 on page 3-65</td>
</tr>
<tr>
<td>Offset</td>
<td>Name</td>
<td>Type</td>
<td>Reset</td>
<td>Width</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------</td>
<td>------</td>
<td>--------------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>0x158</td>
<td>ecc_errc_count_31_00</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.74 ecc_errc_count_31_00 on page 3-66</td>
</tr>
<tr>
<td>0x15C</td>
<td>ecc_errc_count_63_32</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.75 ecc_errc_count_63_32 on page 3-66</td>
</tr>
<tr>
<td>0x160</td>
<td>ecc_errd_count_31_00</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.76 ecc_errd_count_31_00 on page 3-66</td>
</tr>
<tr>
<td>0x164</td>
<td>ecc_errd_count_63_32</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.77 ecc_errd_count_63_32 on page 3-67</td>
</tr>
<tr>
<td>0x168</td>
<td>ram_err_count</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.78 ram_err_count on page 3-67</td>
</tr>
<tr>
<td>0x16C</td>
<td>link_err_count</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.79 link_err_count on page 3-67</td>
</tr>
<tr>
<td>0x170</td>
<td>scrub_control0_next</td>
<td>RW</td>
<td>0x1F000000</td>
<td>32</td>
<td>3.3.80 scrub_control0_next on page 3-67</td>
</tr>
<tr>
<td>0x174</td>
<td>scrub_address_min0_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.81 scrub_address_min0_next on page 3-68</td>
</tr>
<tr>
<td>0x178</td>
<td>scrub_address_max0_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.82 scrub_address_max0_next on page 3-68</td>
</tr>
<tr>
<td>0x180</td>
<td>scrub_control1_next</td>
<td>RW</td>
<td>0x1F000000</td>
<td>32</td>
<td>3.3.83 scrub_control1_next on page 3-68</td>
</tr>
<tr>
<td>0x184</td>
<td>scrub_address_min1_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.84 scrub_address_min1_next on page 3-69</td>
</tr>
<tr>
<td>0x188</td>
<td>scrub_address_max1_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.85 scrub_address_max1_next on page 3-69</td>
</tr>
<tr>
<td>0x190</td>
<td>scrub_control2_next</td>
<td>RW</td>
<td>0x1F000000</td>
<td>32</td>
<td>3.3.86 scrub_control2_next on page 3-69</td>
</tr>
<tr>
<td>0x194</td>
<td>scrub_address_min2_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.87 scrub_address_min2_next on page 3-69</td>
</tr>
<tr>
<td>0x198</td>
<td>scrub_address_max2_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.88 scrub_address_max2_next on page 3-70</td>
</tr>
<tr>
<td>0x1A0</td>
<td>scrub_control3_next</td>
<td>RW</td>
<td>0x1F000000</td>
<td>32</td>
<td>3.3.89 scrub_control3_next on page 3-70</td>
</tr>
<tr>
<td>0x1A4</td>
<td>scrub_address_min3_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.90 scrub_address_min3_next on page 3-70</td>
</tr>
<tr>
<td>0x1A8</td>
<td>scrub_address_max3_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.91 scrub_address_max3_next on page 3-71</td>
</tr>
<tr>
<td>0x1B0</td>
<td>scrub_control4_next</td>
<td>RW</td>
<td>0x1F000000</td>
<td>32</td>
<td>3.3.92 scrub_control4_next on page 3-71</td>
</tr>
<tr>
<td>0x1B4</td>
<td>scrub_address_min4_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.93 scrub_address_min4_next on page 3-71</td>
</tr>
<tr>
<td>0x1B8</td>
<td>scrub_address_max4_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.94 scrub_address_max4_next on page 3-71</td>
</tr>
<tr>
<td>0x1C0</td>
<td>scrub_control5_next</td>
<td>RW</td>
<td>0x1F000000</td>
<td>32</td>
<td>3.3.95 scrub_control5_next on page 3-72</td>
</tr>
<tr>
<td>0x1C4</td>
<td>scrub_address_min5_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.96 scrub_address_min5_next on page 3-72</td>
</tr>
<tr>
<td>0x1C8</td>
<td>scrub_address_max5_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.97 scrub_address_max5_next on page 3-72</td>
</tr>
<tr>
<td>0x1D0</td>
<td>scrub_control6_next</td>
<td>RW</td>
<td>0x1F000000</td>
<td>32</td>
<td>3.3.98 scrub_control6_next on page 3-73</td>
</tr>
<tr>
<td>0x1D4</td>
<td>scrub_address_min6_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.99 scrub_address_min6_next on page 3-73</td>
</tr>
<tr>
<td>0x1D8</td>
<td>scrub_address_max6_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.100 scrub_address_max6_next on page 3-73</td>
</tr>
<tr>
<td>0x1E0</td>
<td>scrub_control7_next</td>
<td>RW</td>
<td>0x1F000000</td>
<td>32</td>
<td>3.3.101 scrub_control7_next on page 3-73</td>
</tr>
<tr>
<td>0x1E4</td>
<td>scrub_address_min7_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.102 scrub_address_min7_next on page 3-74</td>
</tr>
<tr>
<td>0x1E8</td>
<td>scrub_address_max7_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.103 scrub_address_max7_next on page 3-74</td>
</tr>
<tr>
<td>0x1F0</td>
<td>feature_control_next</td>
<td>RW</td>
<td>0x8A000000</td>
<td>32</td>
<td>3.3.104 feature_control_next on page 3-74</td>
</tr>
<tr>
<td>0x1F4</td>
<td>mux_control_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.105 mux_control_next on page 3-75</td>
</tr>
<tr>
<td>0x1F8</td>
<td>rank_remap_control_next</td>
<td>RW</td>
<td>0x76543210</td>
<td>32</td>
<td>3.3.106 rank_remap_control_next on page 3-75</td>
</tr>
<tr>
<td>0x1FC</td>
<td>scrub_control_next</td>
<td>RW</td>
<td>0x00001F00</td>
<td>32</td>
<td>3.3.107 scrub_control_next on page 3-75</td>
</tr>
<tr>
<td>0x200</td>
<td>t_refi_next</td>
<td>RW</td>
<td>0x00090100</td>
<td>32</td>
<td>3.3.108 t_refi_next on page 3-75</td>
</tr>
<tr>
<td>Offset</td>
<td>Name</td>
<td>Type</td>
<td>Reset</td>
<td>Width</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------</td>
<td>------</td>
<td>---------</td>
<td>-------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>0x204</td>
<td>t_rfc_next</td>
<td>RW</td>
<td>0x00008C23</td>
<td>32</td>
<td>3.3.109 t_rfc_next on page 3-76</td>
</tr>
<tr>
<td>0x208</td>
<td>t_mrr_next</td>
<td>RW</td>
<td>0x00000002</td>
<td>32</td>
<td>3.3.110 t_mrr_next on page 3-76</td>
</tr>
<tr>
<td>0x20C</td>
<td>t_mrw_next</td>
<td>RW</td>
<td>0x0000000C</td>
<td>32</td>
<td>3.3.111 t_mrw_next on page 3-76</td>
</tr>
<tr>
<td>0x210</td>
<td>t_rdpden_next</td>
<td>RW</td>
<td>0x0000000A</td>
<td>32</td>
<td>3.3.112 t_rdpden_next on page 3-77</td>
</tr>
<tr>
<td>0x218</td>
<td>t_rcd_next</td>
<td>RW</td>
<td>0x00000005</td>
<td>32</td>
<td>3.3.113 t_rcd_next on page 3-77</td>
</tr>
<tr>
<td>0x21C</td>
<td>t_ras_next</td>
<td>RW</td>
<td>0x0000000E</td>
<td>32</td>
<td>3.3.114 t_ras_next on page 3-77</td>
</tr>
<tr>
<td>0x220</td>
<td>t_rp_next</td>
<td>RW</td>
<td>0x00000005</td>
<td>32</td>
<td>3.3.115 t_rp_next on page 3-78</td>
</tr>
<tr>
<td>0x224</td>
<td>t_rpall_next</td>
<td>RW</td>
<td>0x00000005</td>
<td>32</td>
<td>3.3.116 t_rpall_next on page 3-78</td>
</tr>
<tr>
<td>0x228</td>
<td>t_rrd_next</td>
<td>RW</td>
<td>0x00000404</td>
<td>32</td>
<td>3.3.117 t_rrd_next on page 3-78</td>
</tr>
<tr>
<td>0x22C</td>
<td>t_act_window_next</td>
<td>RW</td>
<td>0x03560014</td>
<td>32</td>
<td>3.3.118 t_act_window_next on page 3-78</td>
</tr>
<tr>
<td>0x234</td>
<td>t_rtr_next</td>
<td>RW</td>
<td>0x00060404</td>
<td>32</td>
<td>3.3.119 t_rtr_next on page 3-79</td>
</tr>
<tr>
<td>0x238</td>
<td>t_rtw_next</td>
<td>RW</td>
<td>0x00060606</td>
<td>32</td>
<td>3.3.120 t_rtw_next on page 3-79</td>
</tr>
<tr>
<td>0x23C</td>
<td>t_rtp_next</td>
<td>RW</td>
<td>0x00000004</td>
<td>32</td>
<td>3.3.121 t_rtp_next on page 3-79</td>
</tr>
<tr>
<td>0x244</td>
<td>t_wr_next</td>
<td>RW</td>
<td>0x00000005</td>
<td>32</td>
<td>3.3.122 t_wr_next on page 3-80</td>
</tr>
<tr>
<td>0x248</td>
<td>t_wtr_next</td>
<td>RW</td>
<td>0x00040505</td>
<td>32</td>
<td>3.3.123 t_wtr_next on page 3-80</td>
</tr>
<tr>
<td>0x24C</td>
<td>t_wtw_next</td>
<td>RW</td>
<td>0x00060404</td>
<td>32</td>
<td>3.3.124 t_wtw_next on page 3-80</td>
</tr>
<tr>
<td>0x254</td>
<td>t_xmpd_next</td>
<td>RW</td>
<td>0x000003FF</td>
<td>32</td>
<td>3.3.125 t_xmpd_next on page 3-81</td>
</tr>
<tr>
<td>0x258</td>
<td>t_ep_next</td>
<td>RW</td>
<td>0x00000002</td>
<td>32</td>
<td>3.3.126 t_ep_next on page 3-81</td>
</tr>
<tr>
<td>0x25C</td>
<td>t_xp_next</td>
<td>RW</td>
<td>0x00060002</td>
<td>32</td>
<td>3.3.127 t_xp_next on page 3-81</td>
</tr>
<tr>
<td>0x260</td>
<td>t_esr_next</td>
<td>RW</td>
<td>0x000000E</td>
<td>32</td>
<td>3.3.128 t_esr_next on page 3-81</td>
</tr>
<tr>
<td>0x264</td>
<td>t_xsr_next</td>
<td>RW</td>
<td>0x05120100</td>
<td>32</td>
<td>3.3.129 t_xsr_next on page 3-82</td>
</tr>
<tr>
<td>0x268</td>
<td>t_esrck_next</td>
<td>RW</td>
<td>0x00000005</td>
<td>32</td>
<td>3.3.130 t_esrck_next on page 3-82</td>
</tr>
<tr>
<td>0x26C</td>
<td>t_ckxsr_next</td>
<td>RW</td>
<td>0x00000001</td>
<td>32</td>
<td>3.3.131 t_ckxsr_next on page 3-82</td>
</tr>
<tr>
<td>0x270</td>
<td>t_cmd_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.132 t_cmd_next on page 3-83</td>
</tr>
<tr>
<td>0x274</td>
<td>t_parity_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.133 t_parity_next on page 3-83</td>
</tr>
<tr>
<td>0x278</td>
<td>t_zqcs_next</td>
<td>RW</td>
<td>0x00000040</td>
<td>32</td>
<td>3.3.134 t_zqcs_next on page 3-83</td>
</tr>
<tr>
<td>0x300</td>
<td>t_rddata_en_next</td>
<td>RW</td>
<td>0x00000001</td>
<td>32</td>
<td>3.3.135 t_rddata_en_next on page 3-84</td>
</tr>
<tr>
<td>0x304</td>
<td>t_phyrdlat_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.136 t_phyrdlat_next on page 3-84</td>
</tr>
<tr>
<td>0x308</td>
<td>t_phywrlat_next</td>
<td>RW</td>
<td>0x00000001</td>
<td>32</td>
<td>3.3.137 t_phywrlat_next on page 3-84</td>
</tr>
<tr>
<td>0x310</td>
<td>rdvl_control_next</td>
<td>RW</td>
<td>0x00001080</td>
<td>32</td>
<td>3.3.138 rdvl_control_next on page 3-84</td>
</tr>
<tr>
<td>0x314</td>
<td>rdvl_mrs_next</td>
<td>RW</td>
<td>0x00000004</td>
<td>32</td>
<td>3.3.139 rdvl_mrs_next on page 3-85</td>
</tr>
<tr>
<td>0x318</td>
<td>rdvl_en_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.140 rdvl_en_next on page 3-85</td>
</tr>
<tr>
<td>0x31C</td>
<td>rdvl_rr_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.141 rdvl_rr_next on page 3-85</td>
</tr>
<tr>
<td>0x320</td>
<td>wrvl_control_next</td>
<td>RW</td>
<td>0x00010000</td>
<td>32</td>
<td>3.3.142 wrvl_control_next on page 3-86</td>
</tr>
<tr>
<td>0x324</td>
<td>wrvl_mrs_next</td>
<td>RW</td>
<td>0x00000086</td>
<td>32</td>
<td>3.3.143 wrvl_mrs_next on page 3-86</td>
</tr>
<tr>
<td>Offset</td>
<td>Name</td>
<td>Type</td>
<td>Reset</td>
<td>Width</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>0x328</td>
<td>t_wrlvl_en_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.144 t_wrlvl_en_next on page 3-86</td>
</tr>
<tr>
<td>0x32C</td>
<td>t_wrlvl_ww_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.145 t_wrlvl_ww_next on page 3-87</td>
</tr>
<tr>
<td>0x348</td>
<td>phy_power_control_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.146 phy_power_control_next on page 3-87</td>
</tr>
<tr>
<td>0x34C</td>
<td>t_lpresp_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.147 t_lpresp_next on page 3-87</td>
</tr>
<tr>
<td>0x350</td>
<td>phy_update_control_next</td>
<td>RW</td>
<td>0xEF000000</td>
<td>32</td>
<td>3.3.148 phy_update_control_next on page 3-88</td>
</tr>
<tr>
<td>0x358</td>
<td>odt_timing_next</td>
<td>RW</td>
<td>0x06000000</td>
<td>32</td>
<td>3.3.149 odt_timing_next on page 3-88</td>
</tr>
<tr>
<td>0x360</td>
<td>odt_wr_control_31_00_next</td>
<td>RW</td>
<td>0x08040201</td>
<td>32</td>
<td>3.3.150 odt_wr_control_31_00_next on page 3-88</td>
</tr>
<tr>
<td>0x364</td>
<td>odt_wr_control_63_32_next</td>
<td>RW</td>
<td>0x80402010</td>
<td>32</td>
<td>3.3.151 odt_wr_control_63_32_next on page 3-88</td>
</tr>
<tr>
<td>0x368</td>
<td>odt_rd_control_31_00_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.152 odt_rd_control_31_00_next on page 3-89</td>
</tr>
<tr>
<td>0x36C</td>
<td>odt_rd_control_63_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.153 odt_rd_control_63_32_next on page 3-89</td>
</tr>
<tr>
<td>0x370</td>
<td>temperature_readout</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.154 temperature_readout on page 3-89</td>
</tr>
<tr>
<td>0x378</td>
<td>training_status</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.155 training_status on page 3-90</td>
</tr>
<tr>
<td>0x37C</td>
<td>update_status</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.156 update_status on page 3-90</td>
</tr>
<tr>
<td>0x380</td>
<td>dq_map_control_15_00_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.157 dq_map_control_15_00_next on page 3-90</td>
</tr>
<tr>
<td>0x384</td>
<td>dq_map_control_31_16_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.158 dq_map_control_31_16_next on page 3-90</td>
</tr>
<tr>
<td>0x388</td>
<td>dq_map_control_47_32_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.159 dq_map_control_47_32_next on page 3-91</td>
</tr>
<tr>
<td>0x38C</td>
<td>dq_map_control_63_48_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.160 dq_map_control_63_48_next on page 3-91</td>
</tr>
<tr>
<td>0x390</td>
<td>dq_map_control_71_64_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.161 dq_map_control_71_64_next on page 3-91</td>
</tr>
<tr>
<td>0x398</td>
<td>rank_status</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.162 rank_status on page 3-92</td>
</tr>
<tr>
<td>0x39C</td>
<td>mode_change_status</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.163 mode_change_status on page 3-92</td>
</tr>
<tr>
<td>0x400</td>
<td>user_status</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.164 user_status on page 3-92</td>
</tr>
<tr>
<td>0x408</td>
<td>user_config0_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.165 user_config0_next on page 3-93</td>
</tr>
<tr>
<td>0x40C</td>
<td>user_config1_next</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.166 user_config1_next on page 3-93</td>
</tr>
<tr>
<td>0x410</td>
<td>user_config2</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.167 user_config2 on page 3-93</td>
</tr>
<tr>
<td>0x414</td>
<td>user_config3</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.168 user_config3 on page 3-93</td>
</tr>
<tr>
<td>0x500</td>
<td>interrupt_control</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.169 interrupt_control on page 3-94</td>
</tr>
<tr>
<td>0x508</td>
<td>interrupt_clr</td>
<td>WO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.170 interrupt_clr on page 3-94</td>
</tr>
<tr>
<td>0x510</td>
<td>interrupt_status</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.171 interrupt_status on page 3-94</td>
</tr>
<tr>
<td>0x518</td>
<td>ram_ecc_errc_int_info_31_00</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.172 ram_ecc_errc_int_info_31_00 on page 3-95</td>
</tr>
<tr>
<td>0x51C</td>
<td>ram_ecc_errc_int_info_63_32</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.173 ram_ecc_errc_int_info_63_32 on page 3-95</td>
</tr>
<tr>
<td>0x520</td>
<td>ram_ecc_errd_int_info_31_00</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.174 ram_ecc_errd_int_info_31_00 on page 3-95</td>
</tr>
<tr>
<td>0x524</td>
<td>ram_ecc_errd_int_info_63_32</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.175 ram_ecc_errd_int_info_63_32 on page 3-95</td>
</tr>
<tr>
<td>0x528</td>
<td>dram_ecc_errc_int_info_31_00</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.176 dram_ecc_errc_int_info_31_00 on page 3-96</td>
</tr>
<tr>
<td>Offset</td>
<td>Name</td>
<td>Type</td>
<td>Reset</td>
<td>Width</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------</td>
<td>--------</td>
<td>----------------</td>
<td>-------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>0x52C</td>
<td>dram_ecc_errc_int_info_63_32</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.177 dram_ecc_errc_int_info_63_32 on page 3-96</td>
</tr>
<tr>
<td>0x530</td>
<td>dram_ecc_errd_int_info_31_00</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.178 dram_ecc_errd_int_info_31_00 on page 3-96</td>
</tr>
<tr>
<td>0x534</td>
<td>dram_ecc_errd_int_info_63_32</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.179 dram_ecc_errd_int_info_63_32 on page 3-97</td>
</tr>
<tr>
<td>0x538</td>
<td>failed_access_int_info_31_00</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.180 failed_access_int_info_31_00 on page 3-97</td>
</tr>
<tr>
<td>0x53C</td>
<td>failed_access_int_info_63_32</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.181 failed_access_int_info_63_32 on page 3-97</td>
</tr>
<tr>
<td>0x540</td>
<td>failed_prog_int_info_31_00</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.182 failed_prog_int_info_31_00 on page 3-97</td>
</tr>
<tr>
<td>0x544</td>
<td>failed_prog_int_info_63_32</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.183 failed_prog_int_info_63_32 on page 3-98</td>
</tr>
<tr>
<td>0x548</td>
<td>link_err_int_info_31_00</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.184 link_err_int_info_31_00 on page 3-98</td>
</tr>
<tr>
<td>0x54C</td>
<td>link_err_int_info_63_32</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.185 link_err_int_info_63_32 on page 3-98</td>
</tr>
<tr>
<td>0x550</td>
<td>arch_fsm_int_info_31_00</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.186 arch_fsm_int_info_31_00 on page 3-99</td>
</tr>
<tr>
<td>0x554</td>
<td>arch_fsm_int_info_63_32</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.187 arch_fsm_int_info_63_32 on page 3-99</td>
</tr>
<tr>
<td>0xE00</td>
<td>integ_cfg</td>
<td>RW</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.188 integ_cfg on page 3-99</td>
</tr>
<tr>
<td>0xE08</td>
<td>integ_outputs</td>
<td>WO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.189 integ_outputs on page 3-99</td>
</tr>
<tr>
<td>0x1010</td>
<td>address_control_now</td>
<td>RO</td>
<td>0x00030202</td>
<td>32</td>
<td>3.3.190 address_control_now on page 3-100</td>
</tr>
<tr>
<td>0x1014</td>
<td>decode_control_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.191 decode_control_now on page 3-100</td>
</tr>
<tr>
<td>0x101C</td>
<td>address_map_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.192 address_map_now on page 3-100</td>
</tr>
<tr>
<td>0x1020</td>
<td>low_power_control_now</td>
<td>RO</td>
<td>0x00000020</td>
<td>32</td>
<td>3.3.193 low_power_control_now on page 3-101</td>
</tr>
<tr>
<td>0x1028</td>
<td>turnaround_control_now</td>
<td>RO</td>
<td>0x08F80F8F</td>
<td>32</td>
<td>3.3.194 turnaround_control_now on page 3-101</td>
</tr>
<tr>
<td>0x102C</td>
<td>hit_turnaround_control_now</td>
<td>RO</td>
<td>0x08909FBF</td>
<td>32</td>
<td>3.3.195 hit_turnaround_control_now on page 3-101</td>
</tr>
<tr>
<td>0x1030</td>
<td>qos_class_control_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.196 qos_class_control_now on page 3-102</td>
</tr>
<tr>
<td>0x1034</td>
<td>escalation_control_now</td>
<td>RO</td>
<td>0x00088F03</td>
<td>32</td>
<td>3.3.197 escalation_control_now on page 3-102</td>
</tr>
<tr>
<td>0x1038</td>
<td>qv_control_31_00_now</td>
<td>RO</td>
<td>0x76543210</td>
<td>32</td>
<td>3.3.198 qv_control_31_00_now on page 3-102</td>
</tr>
<tr>
<td>0x103C</td>
<td>qv_control_63_32_now</td>
<td>RO</td>
<td>0xFEDCBA98</td>
<td>32</td>
<td>3.3.199 qv_control_63_32_now on page 3-102</td>
</tr>
<tr>
<td>0x1040</td>
<td>rt_control_31_00_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.200 rt_control_31_00_now on page 3-103</td>
</tr>
<tr>
<td>0x1044</td>
<td>rt_control_63_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.201 rt_control_63_32_now on page 3-103</td>
</tr>
<tr>
<td>0x1048</td>
<td>timeout_control_now</td>
<td>RO</td>
<td>0x00000001</td>
<td>32</td>
<td>3.3.202 timeout_control_now on page 3-103</td>
</tr>
<tr>
<td>0x104C</td>
<td>credit_control_now</td>
<td>RO</td>
<td>0x000000F0</td>
<td>32</td>
<td>3.3.203 credit_control_now on page 3-104</td>
</tr>
<tr>
<td>0x1050</td>
<td>write_priority_control_31_00_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.204 write_priority_control_31_00_now on page 3-104</td>
</tr>
<tr>
<td>0x1054</td>
<td>write_priority_control_63_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.205 write_priority_control_63_32_now on page 3-104</td>
</tr>
<tr>
<td>0x1060</td>
<td>queue_threshold_control_31_00_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.206 queue_threshold_control_31_00_now on page 3-105</td>
</tr>
<tr>
<td>Offset</td>
<td>Name</td>
<td>Type</td>
<td>Reset</td>
<td>Width</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------------</td>
<td>------</td>
<td>---------------</td>
<td>-------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>0x1064</td>
<td>queue_threshold_control_63_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.207 queue_threshold_control_63_32_now on page 3-105</td>
</tr>
<tr>
<td>0x1078</td>
<td>memory_address_max_31_00_now</td>
<td>RO</td>
<td>0x00000010</td>
<td>32</td>
<td>3.3.208 memory_address_max_31_00_now on page 3-105</td>
</tr>
<tr>
<td>0x107C</td>
<td>memory_address_max_43_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.209 memory_address_max_43_32_now on page 3-105</td>
</tr>
<tr>
<td>0x1080</td>
<td>access_address_min0_31_00_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.210 access_address_min0_31_00_now on page 3-106</td>
</tr>
<tr>
<td>0x1084</td>
<td>access_address_min0_43_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.211 access_address_min0_43_32_now on page 3-106</td>
</tr>
<tr>
<td>0x1088</td>
<td>access_address_max0_31_00_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.212 access_address_max0_31_00_now on page 3-106</td>
</tr>
<tr>
<td>0x108C</td>
<td>access_address_max0_43_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.213 access_address_max0_43_32_now on page 3-107</td>
</tr>
<tr>
<td>0x1090</td>
<td>access_address_min1_31_00_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.214 access_address_min1_31_00_now on page 3-107</td>
</tr>
<tr>
<td>0x1094</td>
<td>access_address_min1_43_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.215 access_address_min1_43_32_now on page 3-107</td>
</tr>
<tr>
<td>0x1098</td>
<td>access_address_max1_31_00_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.216 access_address_max1_31_00_now on page 3-108</td>
</tr>
<tr>
<td>0x109C</td>
<td>access_address_max1_43_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.217 access_address_max1_43_32_now on page 3-108</td>
</tr>
<tr>
<td>0x10A0</td>
<td>access_address_min2_31_00_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.218 access_address_min2_31_00_now on page 3-108</td>
</tr>
<tr>
<td>0x10A4</td>
<td>access_address_min2_43_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.219 access_address_min2_43_32_now on page 3-108</td>
</tr>
<tr>
<td>0x10A8</td>
<td>access_address_max2_31_00_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.220 access_address_max2_31_00_now on page 3-109</td>
</tr>
<tr>
<td>0x10AC</td>
<td>access_address_max2_43_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.221 access_address_max2_43_32_now on page 3-109</td>
</tr>
<tr>
<td>0x10B0</td>
<td>access_address_min3_31_00_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.222 access_address_min3_31_00_now on page 3-109</td>
</tr>
<tr>
<td>0x10B4</td>
<td>access_address_min3_43_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.223 access_address_min3_43_32_now on page 3-110</td>
</tr>
<tr>
<td>0x10B8</td>
<td>access_address_max3_31_00_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.224 access_address_max3_31_00_now on page 3-110</td>
</tr>
<tr>
<td>0x10BC</td>
<td>access_address_max3_43_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.225 access_address_max3_43_32_now on page 3-110</td>
</tr>
<tr>
<td>0x10C0</td>
<td>access_address_min4_31_00_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.226 access_address_min4_31_00_now on page 3-111</td>
</tr>
<tr>
<td>0x10C4</td>
<td>access_address_min4_43_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.227 access_address_min4_43_32_now on page 3-111</td>
</tr>
<tr>
<td>Offset</td>
<td>Name</td>
<td>Type</td>
<td>Reset</td>
<td>Width</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------</td>
<td>------</td>
<td>----------------</td>
<td>-------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>0x10C8</td>
<td>access_address_max4_31_00_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.228 access_address_max4_31_00_now on page 3-111</td>
</tr>
<tr>
<td>0x10CC</td>
<td>access_address_max4_43_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.229 access_address_max4_43_32_now on page 3-111</td>
</tr>
<tr>
<td>0x10D0</td>
<td>access_address_min5_31_00_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.230 access_address_min5_31_00_now on page 3-112</td>
</tr>
<tr>
<td>0x10D4</td>
<td>access_address_min5_43_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.231 access_address_min5_43_32_now on page 3-112</td>
</tr>
<tr>
<td>0x10D8</td>
<td>access_address_max5_31_00_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.232 access_address_max5_31_00_now on page 3-112</td>
</tr>
<tr>
<td>0x10DC</td>
<td>access_address_max5_43_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.233 access_address_max5_43_32_now on page 3-113</td>
</tr>
<tr>
<td>0x10E0</td>
<td>access_address_min6_31_00_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.234 access_address_min6_31_00_now on page 3-113</td>
</tr>
<tr>
<td>0x10E4</td>
<td>access_address_min6_43_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.235 access_address_min6_43_32_now on page 3-113</td>
</tr>
<tr>
<td>0x10E8</td>
<td>access_address_max6_31_00_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.236 access_address_max6_31_00_now on page 3-114</td>
</tr>
<tr>
<td>0x10EC</td>
<td>access_address_max6_43_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.237 access_address_max6_43_32_now on page 3-114</td>
</tr>
<tr>
<td>0x10F0</td>
<td>access_address_min7_31_00_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.238 access_address_min7_31_00_now on page 3-114</td>
</tr>
<tr>
<td>0x10F4</td>
<td>access_address_min7_43_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.239 access_address_min7_43_32_now on page 3-114</td>
</tr>
<tr>
<td>0x10F8</td>
<td>access_address_max7_31_00_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.240 access_address_max7_31_00_now on page 3-115</td>
</tr>
<tr>
<td>0x10FC</td>
<td>access_address_max7_43_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.241 access_address_max7_43_32_now on page 3-115</td>
</tr>
<tr>
<td>0x1110</td>
<td>dci_replay_type_now</td>
<td>RO</td>
<td>0x00000002</td>
<td>32</td>
<td>3.3.242 dci_replay_type_now on page 3-115</td>
</tr>
<tr>
<td>0x1120</td>
<td>refresh_control_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.243 refresh_control_now on page 3-116</td>
</tr>
<tr>
<td>0x1128</td>
<td>memory_type_now</td>
<td>RO</td>
<td>0x00000101</td>
<td>32</td>
<td>3.3.244 memory_type_now on page 3-116</td>
</tr>
<tr>
<td>0x1170</td>
<td>scrub_control0_now</td>
<td>RO</td>
<td>0x1F000000</td>
<td>32</td>
<td>3.3.245 scrub_control0_now on page 3-116</td>
</tr>
<tr>
<td>0x1174</td>
<td>scrub_address_min0_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.246 scrub_address_min0_now on page 3-117</td>
</tr>
<tr>
<td>0x1178</td>
<td>scrub_address_max0_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.247 scrub_address_max0_now on page 3-117</td>
</tr>
<tr>
<td>0x1180</td>
<td>scrub_control1_now</td>
<td>RO</td>
<td>0x1F000000</td>
<td>32</td>
<td>3.3.248 scrub_control1_now on page 3-117</td>
</tr>
<tr>
<td>0x1184</td>
<td>scrub_address_min1_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.249 scrub_address_min1_now on page 3-117</td>
</tr>
<tr>
<td>0x1188</td>
<td>scrub_address_max1_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.250 scrub_address_max1_now on page 3-118</td>
</tr>
<tr>
<td>0x1190</td>
<td>scrub_control2_now</td>
<td>RO</td>
<td>0x1F000000</td>
<td>32</td>
<td>3.3.251 scrub_control2_now on page 3-118</td>
</tr>
<tr>
<td>0x1194</td>
<td>scrub_address_min2_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.252 scrub_address_min2_now on page 3-118</td>
</tr>
<tr>
<td>Offset</td>
<td>Name</td>
<td>Type</td>
<td>Reset</td>
<td>Width</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------</td>
<td>------</td>
<td>-------------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>0x1198</td>
<td>scrub_address_max2_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.253 scrub_address_max2_now on page 3-119</td>
</tr>
<tr>
<td>0x11A0</td>
<td>scrub_control3_now</td>
<td>RO</td>
<td>0x1F000000</td>
<td>32</td>
<td>3.3.254 scrub_control3_now on page 3-119</td>
</tr>
<tr>
<td>0x11A4</td>
<td>scrub_address_min3_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.255 scrub_address_min3_now on page 3-119</td>
</tr>
<tr>
<td>0x11A8</td>
<td>scrub_address_max3_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.256 scrub_address_max3_now on page 3-120</td>
</tr>
<tr>
<td>0x11B0</td>
<td>scrub_control4_now</td>
<td>RO</td>
<td>0x1F000000</td>
<td>32</td>
<td>3.3.257 scrub_control4_now on page 3-120</td>
</tr>
<tr>
<td>0x11B4</td>
<td>scrub_address_min4_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.258 scrub_address_min4_now on page 3-120</td>
</tr>
<tr>
<td>0x11B8</td>
<td>scrub_address_max4_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.259 scrub_address_max4_now on page 3-120</td>
</tr>
<tr>
<td>0x11C0</td>
<td>scrub_control5_now</td>
<td>RO</td>
<td>0x1F000000</td>
<td>32</td>
<td>3.3.260 scrub_control5_now on page 3-121</td>
</tr>
<tr>
<td>0x11C4</td>
<td>scrub_address_min5_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.261 scrub_address_min5_now on page 3-121</td>
</tr>
<tr>
<td>0x11C8</td>
<td>scrub_address_max5_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.262 scrub_address_max5_now on page 3-121</td>
</tr>
<tr>
<td>0x11D0</td>
<td>scrub_control6_now</td>
<td>RO</td>
<td>0x1F000000</td>
<td>32</td>
<td>3.3.263 scrub_control6_now on page 3-122</td>
</tr>
<tr>
<td>0x11D4</td>
<td>scrub_address_min6_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.264 scrub_address_min6_now on page 3-122</td>
</tr>
<tr>
<td>0x11D8</td>
<td>scrub_address_max6_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.265 scrub_address_max6_now on page 3-122</td>
</tr>
<tr>
<td>0x11E0</td>
<td>scrub_control7_now</td>
<td>RO</td>
<td>0x1F000000</td>
<td>32</td>
<td>3.3.266 scrub_control7_now on page 3-123</td>
</tr>
<tr>
<td>0x11E4</td>
<td>scrub_address_min7_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.267 scrub_address_min7_now on page 3-123</td>
</tr>
<tr>
<td>0x11E8</td>
<td>scrub_address_max7_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.268 scrub_address_max7_now on page 3-123</td>
</tr>
<tr>
<td>0x11F0</td>
<td>feature_control_now</td>
<td>RO</td>
<td>0x8A000000</td>
<td>32</td>
<td>3.3.269 feature_control_now on page 3-123</td>
</tr>
<tr>
<td>0x11F4</td>
<td>mux_control_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.270 mux_control_now on page 3-124</td>
</tr>
<tr>
<td>0x11F8</td>
<td>rank_remap_control_now</td>
<td>RO</td>
<td>0x76543210</td>
<td>32</td>
<td>3.3.271 rank_remap_control_now on page 3-124</td>
</tr>
<tr>
<td>0x11FC</td>
<td>scrub_control_now</td>
<td>RO</td>
<td>0x00001F00</td>
<td>32</td>
<td>3.3.272 scrub_control_now on page 3-124</td>
</tr>
<tr>
<td>0x1200</td>
<td>t_refi_now</td>
<td>RO</td>
<td>0x00090100</td>
<td>32</td>
<td>3.3.273 t_refi_now on page 3-125</td>
</tr>
<tr>
<td>0x1204</td>
<td>t_rfc_now</td>
<td>RO</td>
<td>0x000000C3</td>
<td>32</td>
<td>3.3.274 t_rfc_now on page 3-125</td>
</tr>
<tr>
<td>0x1208</td>
<td>t_mrr_now</td>
<td>RO</td>
<td>0x00000002</td>
<td>32</td>
<td>3.3.275 t_mrr_now on page 3-125</td>
</tr>
<tr>
<td>0x120C</td>
<td>t_mrw_now</td>
<td>RO</td>
<td>0x0000000C</td>
<td>32</td>
<td>3.3.276 t_mrw_now on page 3-126</td>
</tr>
<tr>
<td>0x1210</td>
<td>t_rdpden_now</td>
<td>RO</td>
<td>0x0000000A</td>
<td>32</td>
<td>3.3.277 t_rdpden_now on page 3-126</td>
</tr>
<tr>
<td>0x1214</td>
<td>t_rcd_now</td>
<td>RO</td>
<td>0x00000005</td>
<td>32</td>
<td>3.3.278 t_rcd_now on page 3-126</td>
</tr>
<tr>
<td>0x1218</td>
<td>t_ras_now</td>
<td>RO</td>
<td>0x0000000E</td>
<td>32</td>
<td>3.3.279 t_ras_now on page 3-127</td>
</tr>
<tr>
<td>0x1220</td>
<td>t_rp_now</td>
<td>RO</td>
<td>0x00000005</td>
<td>32</td>
<td>3.3.280 t_rp_now on page 3-127</td>
</tr>
<tr>
<td>0x1224</td>
<td>t_rpall_now</td>
<td>RO</td>
<td>0x00000005</td>
<td>32</td>
<td>3.3.281 t_rpall_now on page 3-127</td>
</tr>
<tr>
<td>0x1228</td>
<td>t_rdd_now</td>
<td>RO</td>
<td>0x00000404</td>
<td>32</td>
<td>3.3.282 t_rdd_now on page 3-128</td>
</tr>
<tr>
<td>0x122C</td>
<td>t_act_window_now</td>
<td>RO</td>
<td>0x03560014</td>
<td>32</td>
<td>3.3.283 t_act_window_now on page 3-128</td>
</tr>
<tr>
<td>0x1234</td>
<td>t_rtr_now</td>
<td>RO</td>
<td>0x00006040</td>
<td>32</td>
<td>3.3.284 t_rtr_now on page 3-128</td>
</tr>
<tr>
<td>0x1238</td>
<td>t_rtw_now</td>
<td>RO</td>
<td>0x00006066</td>
<td>32</td>
<td>3.3.285 t_rtw_now on page 3-129</td>
</tr>
<tr>
<td>0x123C</td>
<td>t_rtp_now</td>
<td>RO</td>
<td>0x00000004</td>
<td>32</td>
<td>3.3.286 t_rtp_now on page 3-129</td>
</tr>
<tr>
<td>0x1244</td>
<td>t_wr_now</td>
<td>RO</td>
<td>0x00000005</td>
<td>32</td>
<td>3.3.287 t_wr_now on page 3-129</td>
</tr>
<tr>
<td>Offset</td>
<td>Name</td>
<td>Type</td>
<td>Reset</td>
<td>Width</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------</td>
<td>------</td>
<td>--------------</td>
<td>-------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>0x1248</td>
<td>t_wtr_now</td>
<td>RO</td>
<td>0x00040505</td>
<td>32</td>
<td>3.3.288 t_wtr_now on page 3-130</td>
</tr>
<tr>
<td>0x124C</td>
<td>t_wtw_now</td>
<td>RO</td>
<td>0x00060404</td>
<td>32</td>
<td>3.3.289 t_wtw_now on page 3-130</td>
</tr>
<tr>
<td>0x1254</td>
<td>t_xmpd_now</td>
<td>RO</td>
<td>0x000003FF</td>
<td>32</td>
<td>3.3.290 t_xmpd_now on page 3-130</td>
</tr>
<tr>
<td>0x1258</td>
<td>t_ep_now</td>
<td>RO</td>
<td>0x00000002</td>
<td>32</td>
<td>3.3.291 t_ep_now on page 3-131</td>
</tr>
<tr>
<td>0x125C</td>
<td>t_xp_now</td>
<td>RO</td>
<td>0x00060002</td>
<td>32</td>
<td>3.3.292 t_xp_now on page 3-131</td>
</tr>
<tr>
<td>0x1260</td>
<td>t_esr_now</td>
<td>RO</td>
<td>0x0000000E</td>
<td>32</td>
<td>3.3.293 t_esr_now on page 3-131</td>
</tr>
<tr>
<td>0x1264</td>
<td>t_xsr_now</td>
<td>RO</td>
<td>0x05120100</td>
<td>32</td>
<td>3.3.294 t_xsr_now on page 3-132</td>
</tr>
<tr>
<td>0x1268</td>
<td>t_esrck_now</td>
<td>RO</td>
<td>0x00000005</td>
<td>32</td>
<td>3.3.295 t_esrck_now on page 3-132</td>
</tr>
<tr>
<td>0x126C</td>
<td>t_ckxsr_now</td>
<td>RO</td>
<td>0x00000001</td>
<td>32</td>
<td>3.3.296 t_ckxsr_now on page 3-132</td>
</tr>
<tr>
<td>0x1270</td>
<td>t_cmd_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.297 t_cmd_now on page 3-132</td>
</tr>
<tr>
<td>0x1274</td>
<td>t_parity_now</td>
<td>RO</td>
<td>0x00000900</td>
<td>32</td>
<td>3.3.298 t_parity_now on page 3-133</td>
</tr>
<tr>
<td>0x1278</td>
<td>t_zqcs_now</td>
<td>RO</td>
<td>0x00000040</td>
<td>32</td>
<td>3.3.299 t_zqcs_now on page 3-133</td>
</tr>
<tr>
<td>0x1300</td>
<td>t_rddata_en_now</td>
<td>RO</td>
<td>0x00000001</td>
<td>32</td>
<td>3.3.300 t_rddata_en_now on page 3-133</td>
</tr>
<tr>
<td>0x1304</td>
<td>t_phyrdlat_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.301 t_phyrdlat_now on page 3-134</td>
</tr>
<tr>
<td>0x1308</td>
<td>t_phywrlat_now</td>
<td>RO</td>
<td>0x00000001</td>
<td>32</td>
<td>3.3.302 t_phywrlat_now on page 3-134</td>
</tr>
<tr>
<td>0x1310</td>
<td>rdlvl_control_now</td>
<td>RO</td>
<td>0x00001080</td>
<td>32</td>
<td>3.3.303 rdlvl_control_now on page 3-134</td>
</tr>
<tr>
<td>0x1314</td>
<td>rdlvl_mrs_now</td>
<td>RO</td>
<td>0x00000004</td>
<td>32</td>
<td>3.3.304 rdlvl_mrs_now on page 3-135</td>
</tr>
<tr>
<td>0x1318</td>
<td>t_rdlvl_en_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.305 t_rdlvl_en_now on page 3-135</td>
</tr>
<tr>
<td>0x131C</td>
<td>t_rdlvl_rr_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.306 t_rdlvl_rr_now on page 3-135</td>
</tr>
<tr>
<td>0x1320</td>
<td>wrlvl_control_now</td>
<td>RO</td>
<td>0x00001000</td>
<td>32</td>
<td>3.3.307 wrlvl_control_now on page 3-136</td>
</tr>
<tr>
<td>0x1324</td>
<td>wrlvl_mrs_now</td>
<td>RO</td>
<td>0x00000086</td>
<td>32</td>
<td>3.3.308 wrlvl_mrs_now on page 3-136</td>
</tr>
<tr>
<td>0x1328</td>
<td>t_wrlvl_en_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.309 t_wrlvl_en_now on page 3-136</td>
</tr>
<tr>
<td>0x132C</td>
<td>t_wrlvl_ww_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.310 t_wrlvl_ww_now on page 3-137</td>
</tr>
<tr>
<td>0x1348</td>
<td>phy_power_control_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.311 phy_power_control_now on page 3-137</td>
</tr>
<tr>
<td>0x134C</td>
<td>t_lpresp_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.312 t_lpresp_now on page 3-137</td>
</tr>
<tr>
<td>0x1350</td>
<td>phy_update_control_now</td>
<td>RO</td>
<td>0x0FE0000</td>
<td>32</td>
<td>3.3.313 phy_update_control_now on page 3-138</td>
</tr>
<tr>
<td>0x1358</td>
<td>odt_timing_now</td>
<td>RO</td>
<td>0x06000600</td>
<td>32</td>
<td>3.3.314 odt_timing_now on page 3-138</td>
</tr>
<tr>
<td>0x1360</td>
<td>odt_wr_control_31_00_now</td>
<td>RO</td>
<td>0x08040201</td>
<td>32</td>
<td>3.3.315 odt_wr_control_31_00_now on page 3-138</td>
</tr>
<tr>
<td>0x1364</td>
<td>odt_wr_control_63_32_now</td>
<td>RO</td>
<td>0x86402010</td>
<td>32</td>
<td>3.3.316 odt_wr_control_63_32_now on page 3-139</td>
</tr>
<tr>
<td>0x1368</td>
<td>odt_rd_control_31_00_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.317 odt_rd_control_31_00_now on page 3-139</td>
</tr>
<tr>
<td>0x136C</td>
<td>odt_rd_control_63_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.318 odt_rd_control_63_32_now on page 3-139</td>
</tr>
<tr>
<td>0x1380</td>
<td>dq_map_control_15_00_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.319 dq_map_control_15_00_now on page 3-139</td>
</tr>
<tr>
<td>0x1384</td>
<td>dq_map_control_31_16_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.320 dq_map_control_31_16_now on page 3-140</td>
</tr>
<tr>
<td>0x1388</td>
<td>dq_map_control_47_32_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.321 dq_map_control_47_32_now on page 3-140</td>
</tr>
<tr>
<td>0x138C</td>
<td>dq_map_control_63_48_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.322 dq_map_control_63_48_now on page 3-141</td>
</tr>
<tr>
<td>Offset</td>
<td>Name</td>
<td>Type</td>
<td>Reset</td>
<td>Width</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------</td>
<td>------</td>
<td>---------------</td>
<td>-------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>0x1390</td>
<td>dq_map_control_71_64_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.323 dq_map_control_71_64_now on page 3-141</td>
</tr>
<tr>
<td>0x1408</td>
<td>user_config0_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.324 user_config0_now on page 3-141</td>
</tr>
<tr>
<td>0x140C</td>
<td>user_config1_now</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.325 user_config1_now on page 3-142</td>
</tr>
<tr>
<td>0x1FD0</td>
<td>periph_id_4</td>
<td>RO</td>
<td>0x00000014</td>
<td>32</td>
<td>3.3.326 periph_id_4 on page 3-142</td>
</tr>
<tr>
<td>0x1FE0</td>
<td>periph_id_0</td>
<td>RO</td>
<td>0x00000052</td>
<td>32</td>
<td>3.3.327 periph_id_0 on page 3-142</td>
</tr>
<tr>
<td>0x1FE4</td>
<td>periph_id_1</td>
<td>RO</td>
<td>0x000000B4</td>
<td>32</td>
<td>3.3.328 periph_id_1 on page 3-142</td>
</tr>
<tr>
<td>0x1FE8</td>
<td>periph_id_2</td>
<td>RO</td>
<td>0x0000003B</td>
<td>32</td>
<td>3.3.329 periph_id_2 on page 3-143</td>
</tr>
<tr>
<td>0x1FEC</td>
<td>periph_id_3</td>
<td>RO</td>
<td>0x00000000</td>
<td>32</td>
<td>3.3.330 periph_id_3 on page 3-143</td>
</tr>
<tr>
<td>0x1FF0</td>
<td>component_id_0</td>
<td>RO</td>
<td>0x0000000D</td>
<td>32</td>
<td>3.3.331 component_id_0 on page 3-143</td>
</tr>
<tr>
<td>0x1FF4</td>
<td>component_id_1</td>
<td>RO</td>
<td>0x000000F0</td>
<td>32</td>
<td>3.3.332 component_id_1 on page 3-144</td>
</tr>
<tr>
<td>0x1FF8</td>
<td>component_id_2</td>
<td>RO</td>
<td>0x00000005</td>
<td>32</td>
<td>3.3.333 component_id_2 on page 3-144</td>
</tr>
<tr>
<td>0x1FFC</td>
<td>component_id_3</td>
<td>RO</td>
<td>0x000000B1</td>
<td>32</td>
<td>3.3.334 component_id_3 on page 3-144</td>
</tr>
</tbody>
</table>
3.3 Register descriptions

This section describes the dmc520 registers.

3.2 Register summary on page 3-33 provides cross references to individual registers.

3.3.1 memc_status

Holds the architectural status of the DMC.

The memc_status register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x000
Type Read-only
Reset 0x00000000
Width 32

3.3.2 memc_config

Holds the configuration data for the DMC.

The memc_config register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x004
Type Read-only
Reset 0x00000000
Width 32

3.3.3 memc_cmd

Used to change the architectural state of the DMC, or execute queued manager operations.

The memc_cmd register characteristics are:

Usage constraints
Cannot be read from. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x008
Type Write-only
Reset 0x00000000
Width 32
3.3.4 address_control_next

Configures the DRAM address parameters. Use the DRAM device data sheet or Serial Presence Detect (SPD)-derived values to assist in programming these values.

The address_control_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
- Offset: 0x010
- Type: Read-write
- Reset: 0x00030202
- Width: 32

3.3.5 decode_control_next

Configures how the DRAM address is decoded from the system address. The DRAM address consists of the rank, bank, row address, and the column address.

The decode_control_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
- Offset: 0x014
- Type: Read-write
- Reset: 0x00000000
- Width: 32

3.3.6 format_control

Configures the memory burst access parameters.

The format_control register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in CONFIG or LOW-POWER states.

Configurations
There is only one DMC configuration.

Attributes
- Offset: 0x018
- Type: Read-write
- Reset: 0x22000213
- Width: 32

3.3.7 address_map_next

Configures the system address mapping options.

The address_map_next register characteristics are:
Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x01C
Type Read-write
Reset 0x00000000
Width 32

3.3.8 low_power_control_next

Configures the low-power features of the DMC.

The low_power_control_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x020
Type Read-write
Reset 0x00000020
Width 32

3.3.9 turnaround_control_next

Configures the settings for arbitration between read and write and rank to rank traffic on the DRAM bus.

The turnaround_control_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x028
Type Read-write
Reset 0x0F0F0F0F
Width 32

3.3.10 hit_turnaround_control_next

Configures the settings for preventing starvation of non-hits in the presence of in-row hit streams.

The hit_turnaround_control_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x02C
Type Read-write
Reset 0x08909FBF
3.3.11 qos_class_control_next

Configures the priority class for each QoS encoding.

The qos_class_control_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x030
Type Read-write
Reset 0x00000FC8
Width 32

3.3.12 escalation_control_next

Configures the settings for escalating the priority of entries in the queue.

The escalation_control_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x034
Type Read-write
Reset 0x00008F03
Width 32

3.3.13 qv_control_31_00_next

Configures the priority settings for each QoS encoding.

The qv_control_31_00_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x038
Type Read-write
Reset 0x76543210
Width 32

3.3.14 qv_control_63_32_next

Configures the priority settings for each QoS encoding.

The qv_control_63_32_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.
Configurations

There is only one DMC configuration.

Attributes

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Reset</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x03C</td>
<td>Read-write</td>
<td>0xFEDCBA98</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.15 rt_control_31_00_next

Configures the timeout settings for each QoS encoding.

The rt_control_31_00_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Reset</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x040</td>
<td>Read-write</td>
<td>0x00000000</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.16 rt_control_63_32_next

Configures the timeout settings for each QoS encoding.

The rt_control_63_32_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Reset</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x044</td>
<td>Read-write</td>
<td>0x00000000</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.17 timeout_control_next

Configures the prescaler applied to timeout values.

The timeout_control_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Reset</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x048</td>
<td>Read-write</td>
<td>0x00000001</td>
<td>32</td>
</tr>
</tbody>
</table>
3.3.18 credit_control_next

Configures the settings for preventing starvation of CHI protocol retries.

The credit_control_next register characteristics are:

Usage constraints Can be read from when in ALL states. Can be written to when in ALL states.

Configurations There is only one DMC configuration.

Attributes
Offset 0x04C
Type Read-write
Reset 0x00000F03
Width 32

3.3.19 write_priority_control_31_00_next

Configures the priority settings for write requests within the DMC

The write_priority_control_31_00_next register characteristics are:

Usage constraints Can be read from when in ALL states. Can be written to when in ALL states.

Configurations There is only one DMC configuration.

Attributes
Offset 0x050
Type Read-write
Reset 0x00000000
Width 32

3.3.20 write_priority_control_63_32_next

Configures the priority settings for write requests within the DMC.

The write_priority_control_63_32_next register characteristics are:

Usage constraints Can be read from when in ALL states. Can be written to when in ALL states.

Configurations There is only one DMC configuration.

Attributes
Offset 0x054
Type Read-write
Reset 0x00000000
Width 32

3.3.21 queue_threshold_control_31_00_next

Configures the threshold settings for requests in the DMC.

The queue_threshold_control_31_00_next register characteristics are:

Usage constraints Can be read from when in ALL states. Can be written to when in ALL states.

Configurations There is only one DMC configuration.
Attributes
Offset  0x060
Type    Read-write
Reset   0x00000000
Width   32

3.3.22 queue_threshold_control_63_32_next
Configures the threshold settings for requests in the DMC
The queue_threshold_control_63_32_next register characteristics are:
Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.
Configurations
There is only one DMC configuration.
Attributes
Offset  0x064
Type    Read-write
Reset   0x00000000
Width   32

3.3.23 memory_address_max_31_00_next
Configures the address space control for the DMC default region.
The memory_address_max_31_00_next register characteristics are:
Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.
Configurations
There is only one DMC configuration.
Attributes
Offset  0x078
Type    Read-write
Reset   0x00000010
Width   32

3.3.24 memory_address_max_43_32_next
Configures the address space control for the DMC default region.
The memory_address_max_43_32_next register characteristics are:
Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.
Configurations
There is only one DMC configuration.
Attributes
Offset  0x07C
Type    Read-write
Reset   0x00000000
Width   32
3.3.25 access_address_min0_31_00_next

Configures the address space control for address region 0.

The access_address_min0_31_00_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x080
- Type: Read-write
- Reset: 0x00000000
- Width: 32

3.3.26 access_address_min0_43_32_next

Configures the address space control for address region 0.

The access_address_min0_43_32_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x084
- Type: Read-write
- Reset: 0x00000000
- Width: 32

3.3.27 access_address_max0_31_00_next

Configures the address space control for address region 0.

The access_address_max0_31_00_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x088
- Type: Read-write
- Reset: 0x00000000
- Width: 32

3.3.28 access_address_max0_43_32_next

Configures the address space control for address region 0.

The access_address_max0_43_32_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.
Attributes
   Offset  0x08C
   Type     Read-write
   Reset    0x00000000
   Width    32

3.3.29 access_address_min1_31_00_next

Configures the address space control for address region 1.

The access_address_min1_31_00_next register characteristics are:

Usage constraints
   Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
   There is only one DMC configuration.

Attributes
   Offset  0x090
   Type     Read-write
   Reset    0x00000000
   Width    32

3.3.30 access_address_min1_43_32_next

Configures the address space control for address region 1.

The access_address_min1_43_32_next register characteristics are:

Usage constraints
   Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
   There is only one DMC configuration.

Attributes
   Offset  0x094
   Type     Read-write
   Reset    0x00000000
   Width    32

3.3.31 access_address_max1_31_00_next

Configures the address space control for address region 1.

The access_address_max1_31_00_next register characteristics are:

Usage constraints
   Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
   There is only one DMC configuration.

Attributes
   Offset  0x098
   Type     Read-write
   Reset    0x00000000
   Width    32
3.3.32 access_address_max1_43_32_next

Configures the address space control for address region 1.

The access_address_max1_43_32_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
- Offset: 0x09C
- Type: Read-write
- Reset: 0x00000000
- Width: 32

3.3.33 access_address_min2_31_00_next

Configures the address space control for address region 2.

The access_address_min2_31_00_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
- Offset: 0xA0
- Type: Read-write
- Reset: 0x00000000
- Width: 32

3.3.34 access_address_min2_43_32_next

Configures the address space control for address region 2.

The access_address_min2_43_32_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
- Offset: 0xA4
- Type: Read-write
- Reset: 0x00000000
- Width: 32

3.3.35 access_address_max2_31_00_next

Configures the address space control for address region 2.

The access_address_max2_31_00_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.
Attributes

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x0A8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-write</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

### 3.3.36 access_address_max2_43_32_next

Configures the address space control for address region 2.

The access_address_max2_43_32_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

Attributes

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x0AC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-write</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

### 3.3.37 access_address_min3_31_00_next

Configures the address space control for address region 3.

The access_address_min3_31_00_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

Attributes

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x0B0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-write</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

### 3.3.38 access_address_min3_43_32_next

Configures the address space control for address region 3.

The access_address_min3_43_32_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

Attributes

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x0B4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-write</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>
3.3.39 access_address_max3_31_00_next

Configures the address space control for address region 3.

The access_address_max3_31_00_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset 0x008
- Type Read-write
- Reset 0x00000000
- Width 32

3.3.40 access_address_max3_43_32_next

Configures the address space control for address region 3.

The access_address_max3_43_32_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset 0x0BC
- Type Read-write
- Reset 0x00000000
- Width 32

3.3.41 access_address_min4_31_00_next

Configures the address space control for address region 4.

The access_address_min4_31_00_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset 0x0C0
- Type Read-write
- Reset 0x00000000
- Width 32

3.3.42 access_address_min4_43_32_next

Configures the address space control for address region 4.

The access_address_min4_43_32_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.
Attributes
Offset 0x0C4
Type Read-write
Reset 0x00000000
Width 32

3.3.43 access_address_max4_31_00_next
Configures the address space control for address region 4.
The access_address_max4_31_00_next register characteristics are:
Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.
Configurations
There is only one DMC configuration.
Attributes
Offset 0x0C8
Type Read-write
Reset 0x00000000
Width 32

3.3.44 access_address_max4_43_32_next
Configures the address space control for address region 4.
The access_address_max4_43_32_next register characteristics are:
Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.
Configurations
There is only one DMC configuration.
Attributes
Offset 0x0CC
Type Read-write
Reset 0x00000000
Width 32

3.3.45 access_address_min5_31_00_next
Configures the address space control for address region 5.
The access_address_min5_31_00_next register characteristics are:
Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.
Configurations
There is only one DMC configuration.
Attributes
Offset 0x0D0
Type Read-write
Reset 0x00000000
Width 32
3.3.46 access_address_min5_43_32_next
Configures the address space control for address region 5.
The access_address_min5_43_32_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
- Offset: 0x0D4
- Type: Read-write
- Reset: 0x00000000
- Width: 32

3.3.47 access_address_max5_31_00_next
Configures the address space control for address region 5.
The access_address_max5_31_00_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
- Offset: 0x0D8
- Type: Read-write
- Reset: 0x00000000
- Width: 32

3.3.48 access_address_max5_43_32_next
Configures the address space control for address region 5.
The access_address_max5_43_32_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
- Offset: 0x0DC
- Type: Read-write
- Reset: 0x00000000
- Width: 32

3.3.49 access_address_min6_31_00_next
Configures the address space control for address region 6.
The access_address_min6_31_00_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.
Attributes
Offset 0x0E0
Type Read-write
Reset 0x00000000
Width 32

3.3.50 access_address_min6_43_32_next
Configures the address space control for address region 6.
The access_address_min6_43_32_next register characteristics are:
Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.
Configurations
There is only one DMC configuration.
Attributes
Offset 0x0E4
Type Read-write
Reset 0x00000000
Width 32

3.3.51 access_address_max6_31_00_next
Configures the address space control for address region 6.
The access_address_max6_31_00_next register characteristics are:
Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.
Configurations
There is only one DMC configuration.
Attributes
Offset 0x0E8
Type Read-write
Reset 0x00000000
Width 32

3.3.52 access_address_max6_43_32_next
Configures the address space control for address region 6.
The access_address_max6_43_32_next register characteristics are:
Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.
Configurations
There is only one DMC configuration.
Attributes
Offset 0x0EC
Type Read-write
Reset 0x00000000
Width 32
3.3.53  **access_address_min7_31_00_next**

Configures the address space control for address region 7.

The access_address_min7_31_00_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- **Offset** 0x0f0
- **Type** Read-write
- **Reset** 0x00000000
- **Width** 32

3.3.54  **access_address_min7_43_32_next**

Configures the address space control for address region 7.

The access_address_min7_43_32_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- **Offset** 0x0f4
- **Type** Read-write
- **Reset** 0x00000000
- **Width** 32

3.3.55  **access_address_max7_31_00_next**

Configures the address space control for address region 7.

The access_address_max7_31_00_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- **Offset** 0x0f8
- **Type** Read-write
- **Reset** 0x00000000
- **Width** 32

3.3.56  **access_address_max7_43_32_next**

Configures the address space control for the address region 7.

The access_address_max7_43_32_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.
Attributes
Offset  0x0FC
Type    Read-write
Reset   0x00000000
Width   32

3.3.57 channel_status
Holds the current status of the memory channel.
The channel_status register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes
Offset  0x100
Type    Read-only
Reset   0x00000003
Width   32

3.3.58 direct_addr
Sets the direct command address field for direct commands.
The direct_addr register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in CONFIG, PAUSED or READY states.

Configurations
There is only one DMC configuration.

Attributes
Offset  0x108
Type    Read-write
Reset   0x00000000
Width   32

3.3.59 direct_cmd
Generates direct commands from the manager.
The direct_cmd register characteristics are:

Usage constraints
Cannot be read from. Can be written to when in CONFIG, PAUSED or READY states.

Configurations
There is only one DMC configuration.

Attributes
Offset  0x10C
Type    Write-only
Reset   0x00000000
Width   32
3.3.60  **dci_replay_type_next**

Configures the behavior of the DMC if a DRAM or PHY error is received when executing a direct command.

The `dci_replay_type_next` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
- There is only one DMC configuration.

**Attributes**
- Offset: 0x110
- Type: Read-write
- Reset: 0x00000002
- Width: 32

3.3.61  **dci_strb**

Configures the write data strobe values used during direct_cmd WRITE operations.

The `dci_strb` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Can be written to when in CONFIG, PAUSED or READY states.

**Configurations**
- There is only one DMC configuration.

**Attributes**
- Offset: 0x118
- Type: Read-write
- Reset: 0x0000000F
- Width: 32

3.3.62  **dci_data**

Reading from this register location returns read data received as a result of a READ command. Writing to this register location sets the data to be used for direct_cmd WRITE commands. You must read or write once for each 32-bit data word of a DRAM burst.

The `dci_data` register characteristics are:

**Usage constraints**
- Can be read from when in CONFIG, PAUSED or READY states. Can be written to when in CONFIG, PAUSED or READY states.

**Configurations**
- There is only one DMC configuration.

**Attributes**
- Offset: 0x11C
- Type: Read-write
- Reset: 0x00000000
- Width: 32

3.3.63  **refresh_control_next**

Configures the type of refresh commands issued by the DMC.

The `refresh_control_next` register characteristics are:
Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
  Offset  0x120
  Type    Read-write
  Reset   0x00000000
  Width   32

3.3.64 memory_type_next
Configures the DMC for the attached memory type.
The memory_type_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
  Offset  0x128
  Type    Read-write
  Reset   0x00000101
  Width   32

3.3.65 feature_config
Control register for DMC features.
The feature_config register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in CONFIG or LOW-POWER states.

Configurations
There is only one DMC configuration.

Attributes
  Offset  0x130
  Type    Read-write
  Reset   0x000000F0
  Width   32

3.3.66 nibble_failed_031_000
Used to inform the DMC that a particular nibble has failed.
The nibble_failed_031_000 register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
  Offset  0x138
  Type    Read-write
3.3.67 nibble_failed_063_032
Used to inform the DMC that a particular nibble has failed.
The nibble_failed_063_032 register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.
Configurations
There is only one DMC configuration.
Attributes

- Offset: 0x13C
- Type: Read-write
- Reset: 0x00000000
- Width: 32

3.3.68 nibble_failed_095_064
Used to inform the DMC that a particular nibble has failed.
The nibble_failed_095_064 register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.
Configurations
There is only one DMC configuration.
Attributes

- Offset: 0x140
- Type: Read-write
- Reset: 0x00000000
- Width: 32

3.3.69 nibble_failed_127_096
Used to inform the DMC that a particular nibble has failed.
The nibble_failed_127_096 register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.
Configurations
There is only one DMC configuration.
Attributes

- Offset: 0x144
- Type: Read-write
- Reset: 0x00000000
- Width: 32

3.3.70 queue_allocate_control_031_000
Used to inform the DMC that a particular queue (RAM) entry has failed, where 0 means failed and not included for allocation.
The queue_allocate_control_031_000 register characteristics are:
Usage constraints
Can be read from when in ALL states. Can be written to when in CONFIG or LOW-POWER states.

Configurations
There is only one DMC configuration.

Attributes
Offset  0x148
Type    Read-write
Reset   0xFFFFFFFF
Width   32

3.3.71 queue_allocate_control_063_032
Configures the DMC to not allocate particular queue entries (one bit per entry), for example to avoid using faulty internal RAM locations.

The queue_allocate_control_063_032 register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in CONFIG or LOW-POWER states.

Configurations
There is only one DMC configuration.

Attributes
Offset  0x14C
Type    Read-write
Reset   0xFFFFFFFF
Width   32

3.3.72 queue_allocate_control_095_064
Configures the DMC to not allocate particular queue entries (one bit per entry), for example to avoid using faulty internal RAM locations.

The queue_allocate_control_095_064 register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in CONFIG or LOW-POWER states.

Configurations
There is only one DMC configuration.

Attributes
Offset  0x150
Type    Read-write
Reset   0xFFFFFFFF
Width   32

3.3.73 queue_allocate_control_127_096
Configures the DMC to not allocate particular queue entries (one bit per entry), for example to avoid using faulty internal RAM locations.

The queue_allocate_control_127_096 register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in CONFIG or LOW-POWER states.
3.3.74  **ecc_errc_count_31_00**

Counter register for the DRAM ECC functionality.

The ecc_errc_count_31_00 register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Attributes**
- Offset: 0x154
- Type: Read-write
- Reset: 0xFFFFFFFF
- Width: 32

3.3.75  **ecc_errc_count_63_32**

Counter register for the DRAM ECC functionality.

The ecc_errc_count_63_32 register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Attributes**
- Offset: 0x158
- Type: Read-write
- Reset: 0x00000000
- Width: 32

3.3.76  **ecc_errd_count_31_00**

Counter register for the DRAM ECC functionality.

The ecc_errd_count_31_00 register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Attributes**
- Offset: 0x160
- Type: Read-write
- Reset: 0x00000000
- Width: 32
### 3.3.77 **ecc_errd_count_63_32**
Counter register for the DRAM ECC functionality.

The `ecc_errd_count_63_32` register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x164
- Type: Read-write
- Reset: 0x00000000
- Width: 32

### 3.3.78 **ram_err_count**
Counter register for the RAM ECC functionality.

The `ram_err_count` register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x168
- Type: Read-write
- Reset: 0x00000000
- Width: 32

### 3.3.79 **link_err_count**
Counter register for link errors. The counter increments on detection of a new link error (`dfi_alert_n` or `dfi_err`).

The `link_err_count` register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x16c
- Type: Read-write
- Reset: 0x00000000
- Width: 32

### 3.3.80 **scrub_control0_next**
Scrub engine channel control register.

The `scrub_control0_next` register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.
Configurations
There is only one DMC configuration.

Attributes
Offset 0x170
Type Read-write
Reset 0x1F000000
Width 32

3.3.81 scrub_address_min0_next
Configures the address space control for the scrub engine channel.
The scrub_address_min0_next register characteristics are:
Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.
Configurations
There is only one DMC configuration.
Attributes
Offset 0x174
Type Read-write
Reset 0x00000000
Width 32

3.3.82 scrub_address_max0_next
Configures the address space control for the scrub engine channel.
The scrub_address_max0_next register characteristics are:
Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.
Configurations
There is only one DMC configuration.
Attributes
Offset 0x178
Type Read-write
Reset 0x00000000
Width 32

3.3.83 scrub_control1_next
Scrub engine channel control register.
The scrub_control1_next register characteristics are:
Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.
Configurations
There is only one DMC configuration.
Attributes
Offset 0x180
Type Read-write
Reset 0x1F000000
Width 32
3.3.84 **scrub_address_min1_next**

Configures the address space control for the scrub engine channel.

The `scrub_address_min1_next` register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- **Offset**: 0x184
- **Type**: Read-write
- **Reset**: 0x00000000
- **Width**: 32

3.3.85 **scrub_address_max1_next**

Configures the address space control for the scrub engine channel.

The `scrub_address_max1_next` register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- **Offset**: 0x188
- **Type**: Read-write
- **Reset**: 0x00000000
- **Width**: 32

3.3.86 **scrub_control2_next**

Scrub engine channel control register.

The `scrub_control2_next` register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- **Offset**: 0x190
- **Type**: Read-write
- **Reset**: 0x1F000000
- **Width**: 32

3.3.87 **scrub_address_min2_next**

Configures the address space control for the scrub engine channel.

The `scrub_address_min2_next` register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.
Attributes
  Offset  0x194
  Type    Read-write
  Reset   0x00000000
  Width   32

3.3.88 scrub_address_max2_next

Configures the address space control for the scrub engine channel.

The scrub_address_max2_next register characteristics are:

Usage constraints
  Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
  There is only one DMC configuration.

Attributes
  Offset  0x198
  Type    Read-write
  Reset   0x00000000
  Width   32

3.3.89 scrub_control3_next

Scrub engine channel control register.

The scrub_control3_next register characteristics are:

Usage constraints
  Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
  There is only one DMC configuration.

Attributes
  Offset  0x1A0
  Type    Read-write
  Reset   0x1F000000
  Width   32

3.3.90 scrub_address_min3_next

Configures the address space control for the scrub engine channel.

The scrub_address_min3_next register characteristics are:

Usage constraints
  Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
  There is only one DMC configuration.

Attributes
  Offset  0x1A4
  Type    Read-write
  Reset   0x00000000
  Width   32
3.3.91  **scrub_address_max3_next**

Configures the address space control for the scrub engine channel.

The `scrub_address_max3_next` register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Reset</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x1A8</td>
<td>Read-write</td>
<td>0x00000000</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.92  **scrub_control4_next**

Scrub engine channel control register.

The `scrub_control4_next` register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Reset</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x1B0</td>
<td>Read-write</td>
<td>0x1F000000</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.93  **scrub_address_min4_next**

Configures the address space control for the scrub engine channel.

The `scrub_address_min4_next` register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Reset</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x1B4</td>
<td>Read-write</td>
<td>0x00000000</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.94  **scrub_address_max4_next**

Configures the address space control for the scrub engine channel.

The `scrub_address_max4_next` register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.
3.3.95 scrub_control5_next

Scrub engine channel control register.

The scrub_control5_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x1B8
Type Read-write
Reset 0x00000000
Width 32

3.3.96 scrub_address_min5_next

Configures the address space control for the scrub engine channel.

The scrub_address_min5_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x1C0
Type Read-write
Reset 0x1F000000
Width 32

3.3.97 scrub_address_max5_next

Configures the address space control for the scrub engine channel.

The scrub_address_max5_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x1C4
Type Read-write
Reset 0x00000000
Width 32
3.3.98 **scrub_control6_next**

Scrub engine channel control register.

The scrub_control6_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- **Offset**: 0x1D0
- **Type**: Read-write
- **Reset**: 0x1F000000
- **Width**: 32

3.3.99 **scrub_address_min6_next**

Configures the address space control for the scrub engine channel.

The scrub_address_min6_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- **Offset**: 0x1D4
- **Type**: Read-write
- **Reset**: 0x00000000
- **Width**: 32

3.3.100 **scrub_address_max6_next**

Configures the address space control for the scrub engine channel.

The scrub_address_max6_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- **Offset**: 0x1D8
- **Type**: Read-write
- **Reset**: 0x00000000
- **Width**: 32

3.3.101 **scrub_control7_next**

Scrub engine channel control register.

The scrub_control7_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.
Attributes
Offset  0x1E0
Type    Read-write
Reset   0x1F000000
Width   32

3.3.102 scrub_address_min7_next
Configures the address space control for the scrub engine channel.
The scrub_address_min7_next register characteristics are:
Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.
Configurations
There is only one DMC configuration.
Attributes
Offset  0x1E4
Type    Read-write
Reset   0x00000000
Width   32

3.3.103 scrub_address_max7_next
Configures the address space control for the scrub engine channel.
The scrub_address_max7_next register characteristics are:
Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.
Configurations
There is only one DMC configuration.
Attributes
Offset  0x1E8
Type    Read-write
Reset   0x00000000
Width   32

3.3.104 feature_control_next
Control register for DMC features.
The feature_control_next register characteristics are:
Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.
Configurations
There is only one DMC configuration.
Attributes
Offset  0x1F0
Type    Read-write
Reset   0x0A000000
Width   32
3.3.105 mux_control_next

Control muxing options for the DMC.

The mux_control_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- **Offset**: 0x1F4
- **Type**: Read-write
- **Reset**: 0x00000000
- **Width**: 32

3.3.106 rank_remap_control_next

Control register for rank remap.

The rank_remap_control_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- **Offset**: 0x1F8
- **Type**: Read-write
- **Reset**: 0x76543210
- **Width**: 32

3.3.107 scrub_control_next

Scrub engine channel control register.

The scrub_control_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- **Offset**: 0x1FC
- **Type**: Read-write
- **Reset**: 0x00001F00
- **Width**: 32

3.3.108 t_refi_next

Configures the refresh interval timing parameter. It must be programmed to the device average all-bank AUTOREFRESH interval, divided by 8.

The t_refi_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.
3.3.109 **t_rfc_next**

Configures the tRFC timing parameter. This determines the delay applied after an AUTOREFRESH command before any other command is issued to the same rank.

The t_rfc_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**

- **Offset**: 0x200
- **Type**: Read-write
- **Reset**: 0x00090100
- **Width**: 32

3.3.110 **t_mrr_next**

Configures the tMRR timing parameter. This determines the Mode Register Read (including Multi-Purpose Register Reads) command delay before any other command is issued to the same rank.

The t_mrr_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**

- **Offset**: 0x204
- **Type**: Read-write
- **Reset**: 0x00008C23
- **Width**: 32

3.3.111 **t_mrw_next**

Configures the tMRW timing parameter. This determines the delay applied after a Mode Register Write (including Multi-Purpose Register Writes) command before any other command is issued to the same rank.

The t_mrw_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**

- **Offset**: 0x208
- **Type**: Read-write
- **Reset**: 0x00000002
- **Width**: 32
3.3.112  **t_rdpden_next**

Configures the tRDPDEN timing parameter. This determines the delay applied after a Read command before a power down command can be issued to the same rank.

The t_rdpden_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Reset</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x210</td>
<td>Read-write</td>
<td>0x0000000C</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.113  **t_rcd_next**

Configures the tRCD timing parameter. This determines the delay applied after an ACTIVATE command before a READ or WRITE command is issued to the same bank.

The t_rcd_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Reset</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x218</td>
<td>Read-write</td>
<td>0x00000005</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.114  **t_ras_next**

Configures the tRAS timing parameter. This determines the delay applied after an ACTIVATE command before a PRECHARGE command is issued to the same bank.

The t_ras_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Reset</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x21C</td>
<td>Read-write</td>
<td>0x0000000E</td>
<td>32</td>
</tr>
</tbody>
</table>
3.3.115  t_rp_next

Configures the tRP timing parameter. This determines the delay applied after a PRECHARGE command before any other command is issued to the same bank.

The t_rp_next register characteristics are:

Usage constraints
  Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
  There is only one DMC configuration.

Attributes
  Offset  0x220
  Type    Read-write
  Reset   0x00000005
  Width   32

3.3.116  t_rpall_next

Configures the tRPALL timing parameter. This determines the delay applied after a PRECHARGEALL command before any other command is issued to the same rank.

The t_rpall_next register characteristics are:

Usage constraints
  Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
  There is only one DMC configuration.

Attributes
  Offset  0x224
  Type    Read-write
  Reset   0x00000005
  Width   32

3.3.117  t_rrd_next

Configures the tRRD timing parameter. This determines the delay applied after an ACTIVATE command before another ACTIVATE command is issued to the same rank. The _l and _s fields apply to the same bank group, and a different bank group, respectively, as described in the DDR4 specification.

The t_rrd_next register characteristics are:

Usage constraints
  Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
  There is only one DMC configuration.

Attributes
  Offset  0x228
  Type    Read-write
  Reset   0x00000404
  Width   32

3.3.118  t_act_window_next

Configures the tFAW and tMAWi timing parameters.

The t_act_window_next register characteristics are:
Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x22C
Type Read-write
Reset 0x03560014
Width 32

3.3.119  t_rtr_next

Configures the read-to-read timing parameter. This determines the READ to READ command delay applied between reads to the same chip, other bank group (t_rtr_s), same chip, same bank group (t_rtr_l), and different chip-selects (t_rtr_cs).

The t_rtr_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x234
Type Read-write
Reset 0x00060404
Width 32

3.3.120  t_rtw_next

Configures the read-to-write timing parameter. This determines the READ to WRITE command delay applied between issued commands to the same chip, other bank group (t_rtw_s), same chip, same bank group (t_trw_l), and other chip-selects (t_rtw_cs).

The t_rtw_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x238
Type Read-write
Reset 0x00060606
Width 32

3.3.121  t_rtp_next

Configures the read-to-precharge timing parameter. This determines the READ to PRECHARGE command delay applied between issued commands to the same bank.

The t_rtp_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.
3.3.122 t_wr_next

Configures the tWR timing parameter. This determines the write recovery time and is used as the delay applied between the issue of a WRITE command and subsequent commands, other than WRITEs, to the same bank.

The t_wr_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x23C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-write</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000004</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.123 t_wtr_next

Configures the write-to-read timing parameter, for both same chip, other bank group (tWTR_s), same chip, same bank group (t_WTR_l), and alternate chip (tWTR_cs).

The t_wtr_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x244</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-write</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000005</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.124 t_wtw_next

Configures the write-to-write timing parameter for same chip, other bank group (t_wtw_s), same chip, same bank group (t_wtw_l), alternate chip (t_wtw_cs) writes.

The t_wtw_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x24C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-write</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00060404</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>
3.3.125  t_xmpd_next

Configures the command delay between exiting Maximum Power Down and a subsequent command to that rank.

The t_xmpd_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x254
Type Read-write
Reset 0x000003FF
Width 32

3.3.126  t_ep_next

Configures the enter power-down timing parameter. This parameter is applied between the issue of an active or precharge power down request and subsequent commands to the same rank.

The t_ep_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x258
Type Read-write
Reset 0x00000002
Width 32

3.3.127  t_xp_next

Configures the exit power-down timing parameter for operations that do not require a DLL (tXP), and those that do (tXPDLL).

The t_xp_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x25C
Type Read-write
Reset 0x00060002
Width 32

3.3.128  t_esr_next

Configures the enter self-refresh timing parameter. This parameter is applied between issue of an enter self-refresh request and subsequent commands to the same rank.
The t_esr_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset 0x260
- Type Read-write
- Reset 0x0000000E
- Width 32

3.3.129 t_xsr_next

Configures the exit self-refresh timing parameter. This parameter is applied between the issue of an exit self-refresh request and subsequent commands to the same rank.

The t_xsr_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset 0x264
- Type Read-write
- Reset 0x05120100
- Width 32

3.3.130 t_esrck_next

Configures the delay between entering self-refresh and disabling the DRAM clock. This parameter is applied when stopping the clock when in self-refresh and when in a maximum power-down state.

The t_esrck_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset 0x268
- Type Read-write
- Reset 0x00000005
- Width 32

3.3.131 t_ckxsr_next

Configures the delay between DRAM clock enable and exiting self-refresh. This parameter is applied when re-instating the clock when in self-refresh and when in a maximum power-down state.

The t_ckxsr_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.
Attributes
Offset 0x26C
Type Read-write
Reset 0x00000001
Width 32

3.3.132 t_cmd_next
Configures command signaling timing.
The t_cmd_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x270
Type Read-write
Reset 0x00000000
Width 32

3.3.133 t_parity_next
Parity latencies t_parinlat and t_completion.
The t_parity_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x274
Type Read-write
Reset 0x00000000
Width 32

3.3.134 t_zqcs_next
Configures the delay to apply following a ZQC-Short calibration command.
The t_zqcs_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x278
Type Read-write
Reset 0x00000040
Width 32
3.3.135  t_rddata_en_next

Determines the time between a READ command commencing on the DFI interface, and the assertion of
the dfi_read_en signal.

The t_rddata_en_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
  Offset  0x300
  Type    Read-write
  Reset   0x00000001
  Width   32

3.3.136  t_phyrdlat_next

Determines the maximum possible time between the assertion of the dfi_read_en signal, and the
assertion of the dfi_rddata_valid signal by the PHY.

The t_phyrdlat_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
  Offset  0x304
  Type    Read-write
  Reset   0x00000000
  Width   32

3.3.137  t_phywrlat_next

Determines the time between a WRITE command commencing on the DFI interface, and the assertion of
the dfi_wrdata_en, dfi_wrdata_cs and dfi_wrdata signals.

The t_phywrlat_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
  Offset  0x308
  Type    Read-write
  Reset   0x00000001
  Width   32

3.3.138  rdlvl_control_next

Determines the DMC behavior during read training operations. See the PHY training interface section of
the Integration Manual for more details on PHY training.

The rdlvl_control_next register characteristics are:
Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
<table>
<thead>
<tr>
<th>Offset</th>
<th>0x310</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-write</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00001080</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.139 rdlvl_mrs_next
Determines the Mode Register command to use to place the DRAM into a training mode for read training, when enabled by the rdlvl_control register. See the PHY interface section of the Integration Manual for more information on PHY training.

The rdlvl_mrs_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
<table>
<thead>
<tr>
<th>Offset</th>
<th>0x314</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-write</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000004</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.140 t_rdlvl_en_next
Configures the t_rdlvl_en timing parameter. This specifies the cycle delay between asserting dfi_rdlvl_en and the first training command, and also the cycle delay between deasserting dfi_rdlvl_en and performing any subsequent command. It also specifies the minimum delay between training commands and refreshes during training.

The t_rdlvl_en_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
<table>
<thead>
<tr>
<th>Offset</th>
<th>0x318</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-write</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.141 t_rdlvl_rr_next
Configures the t_rdlvl_rr timing parameter. This specifies the cycle delay between training commands. It also specifies the minimum delay between the last training command and deasserting dfi_rdlvl_en after observing dfi_rdlvl_resp.

The t_rdlvl_rr_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.
3.3.142  

**wrlvl_control_next**

Determines the DMC behavior during write training operations. See the PHY training interface section of the Integration Manual for more information on PHY training.

The `wrlvl_control_next` register characteristics are:

**Usage constraints**

Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**

There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x31C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-write</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.143  

**wrlvl_mrs_next**

Determines the Mode Register command that the DMC must use to put the DRAM into a training mode for write leveling. You enable this function with the `wrlvl_control` Register. See the PHY training interface section of the Integration Manual for more information.

The `wrlvl_mrs_next` register characteristics are:

**Usage constraints**

Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**

There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x320</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-write</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00001000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.144  

**t_wrlvl_en_next**

Configures the `t_wrlvl_en` timing parameter. Specifies the cycle delay between asserting ODT for training and asserting `dfi_wrlvl_en`, the delay between asserting `dfi_wrlvl_en` and the first training command, the delay between deasserting `dfi_wrlvl_en` and de-asserting ODT, and deasserting ODT to any subsequent command. It is also used between ODT transitions and refreshes generated during training.

The `t_wrlvl_en_next` register characteristics are:

**Usage constraints**

Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**

There is only one DMC configuration.
Attributes
- Offset: 0x328
- Type: Read-write
- Reset: 0x00000000
- Width: 32

### 3.3.145 t_wrlvl_ww_next
Configures the t_wrlvl_ww timing parameter. Specifies the cycle delay between training commands. Also specifies the minimum delay between the last training command and de-asserting dfi_wrlvl_en on observing dfi_wrlvl_resp.

The t_wrlvl_ww_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

Attributes
- Offset: 0x32C
- Type: Read-write
- Reset: 0x00000000
- Width: 32

### 3.3.146 phy_power_control_next
Configures the low-power requests made to the PHY for the different channel states.

The phy_power_control_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

Attributes
- Offset: 0x348
- Type: Read-write
- Reset: 0x00000000
- Width: 32

### 3.3.147 t_lresp_next
Configures the minimum cycle delay to apply for PHY low-power handshakes.

The t_lresp_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

Attributes
- Offset: 0x34C
- Type: Read-write
- Reset: 0x00000000
- Width: 32
3.3.148 phy_update_control_next

Configures the update mechanism to use in response to PHY training requests.

The phy_update_control_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x350
Type Read-write
Reset 0x0FE00000
Width 32

3.3.149 odt_timing_next

Configures the ODT on and off timing.

The odt_timing_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x358
Type Read-write
Reset 0x06000600
Width 32

3.3.150 odt_wr_control_31_00_next

Configures the ODT on and off settings for active and inactive ranks during writes.

The odt_wr_control_31_00_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x360
Type Read-write
Reset 0x08040201
Width 32

3.3.151 odt_wr_control_63_32_next

Configures the ODT on and off settings for active and inactive ranks during writes.

The odt_wr_control_63_32_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.
Attributes

Offset 0x364
Type Read-write
Reset 0x80402010
Width 32

3.3.152 odt_rd_control_31_00_next

Configures the ODT on and off settings for active and inactive ranks during reads.
The odt_rd_control_31_00_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes

Offset 0x368
Type Read-write
Reset 0x00000000
Width 32

3.3.153 odt_rd_control_63_32_next

Configures the ODT on and off settings for active and inactive ranks during reads.
The odt_rd_control_63_32_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes

Offset 0x36C
Type Read-write
Reset 0x00000000
Width 32

3.3.154 temperature_readout

Holds the status of the temperature information. Reading the register returns the current temperature from the most recent automated temperature poll.
The temperature_readout register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes

Offset 0x370
Type Read-only
Reset 0x00000000
Width 32
3.3.155 training_status

Shows information relating to the training request status of the DMC.

The training_status register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x378
Type Read-only
Reset 0x00000000
Width 32

3.3.156 update_status

Shows information relating to the update request status of the DMC.

The update_status register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x37C
Type Read-only
Reset 0x00000000
Width 32

3.3.157 dq_map_control_15_00_next

Controls the DQ mapping compensation applied for CRC calculation. For each nibble of the DQ bus, the DIMM SPD defines a DQ Map Index to define the bit connectivity to each rank. Program the DQ Map Index retrieved from the SPD for each nibble into the corresponding register in the DMC for correct CRC operation.

The dq_map_control_15_00_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x380
Type Read-write
Reset 0x00000000
Width 32

3.3.158 dq_map_control_31_16_next

Controls the DQ mapping compensation applied for CRC calculation. For each nibble of the DQ bus, the DIMM SPD defines a DQ Map Index to define the bit connectivity to each rank. Program the DQ Map Index retrieved from the SPD for each nibble into the corresponding register in the DMC for correct CRC operation.
The dq_map_control_31_16_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x384
- Type: Read-write
- Reset: 0x00000000
- Width: 32

### 3.3.159 dq_map_control_47_32_next

Controls the DQ mapping compensation applied for CRC calculation. For each nibble of the DQ bus, the DIMM SPD defines a DQ Map Index to define the bit connectivity to each rank. Program the DQ Map Index retrieved from the SPD for each nibble into the corresponding register in the DMC for correct CRC operation.

The dq_map_control_47_32_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x388
- Type: Read-write
- Reset: 0x00000000
- Width: 32

### 3.3.160 dq_map_control_63_48_next

Controls the DQ mapping compensation applied for CRC calculation. For each nibble of the DQ bus, the DIMM SPD defines a DQ Map Index to define the bit connectivity to each rank. Program the DQ Map Index retrieved from the SPD for each nibble into the corresponding register in the DMC for correct CRC operation.

The dq_map_control_63_48_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x38C
- Type: Read-write
- Reset: 0x00000000
- Width: 32

### 3.3.161 dq_map_control_71_64_next

Controls the DQ mapping compensation applied for CRC calculation. For each nibble of the DQ bus, the DIMM SPD defines a DQ Map Index to define the bit connectivity to each rank. Program the DQ Map Index retrieved from the SPD for DIMM Check Bits bus into this register in the DMC for correct CRC operation.
The dq_map_control_71_64_next register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Can be written to when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x390
- Type: Read-write
- Reset: 0x00000000
- Width: 32

### 3.3.162 rank_status

Shows the current status of geardown, MPD and CAL.

The rank_status register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be changed.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x398
- Type: Read-only
- Reset: 0x00000000
- Width: 32

### 3.3.163 mode_change_status

Shows the current status of the sequence that is currently being processed.

The mode_change_status register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be changed.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x39C
- Type: Read-only
- Reset: 0x00000000
- Width: 32

### 3.3.164 user_status

Shows the value of the input user_status signals.

The user_status register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be changed.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x400
3.3.165 user_config0_next

Drives the output user_config0 signal.

The user_config0_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x408
Type Read-write
Reset 0x00000000
Width 32

3.3.166 user_config1_next

Drives the output user_config1 signal.

The user_config1_next register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x40C
Type Read-write
Reset 0x00000000
Width 32

3.3.167 user_config2

Drives the output user_config2 signal.

The user_config2 register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in ALL states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x410
Type Read-write
Reset 0x00000000
Width 32

3.3.168 user_config3

Drives the output user_config3 signal.

The user_config3 register characteristics are:
Usage constraints  Can be read from when in ALL states. Can be written to when in ALL states.
Configurations  There is only one DMC configuration.
Attributes  
  Offset   0x414  
  Type      Read-write  
  Reset    0x00000000  
  Width    32

3.3.169 interrupt_control
Configures interrupt behavior.
The interrupt_control register characteristics are:
Usage constraints  Can be read from when in ALL states. Can be written to when in ALL states.
Configurations  There is only one DMC configuration.
Attributes  
  Offset   0x500  
  Type      Read-write  
  Reset    0x00000000  
  Width    32

3.3.170 interrupt_clr
Clear register for interrupts.
The interrupt_clr register characteristics are:
Usage constraints  Cannot be read from. Can be written to when in ALL states.
Configurations  There is only one DMC configuration.
Attributes  
  Offset   0x508  
  Type      Write-only  
  Reset    0x00000000  
  Width    32

3.3.171 interrupt_status
Status register for interrupts (pre-mask).
The interrupt_status register characteristics are:
Usage constraints  Can be read from when in ALL states. Cannot be changed.
Configurations  There is only one DMC configuration.
Attributes  
  Offset   0x510  
  Type      Read-only  
  Reset    0x00000000
3.3.172  ram_ecc_errc_int_info_31_00
Shows information relating to the interrupt
The ram_ecc_errc_int_info_31_00 register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x518
Type Read-only
Reset 0x00000000
Width 32

3.3.173  ram_ecc_errc_int_info_63_32
Shows information relating to the interrupt
The ram_ecc_errc_int_info_63_32 register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x51C
Type Read-only
Reset 0x00000000
Width 32

3.3.174  ram_ecc_errd_int_info_31_00
Shows information relating to the interrupt
The ram_ecc_errd_int_info_31_00 register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x520
Type Read-only
Reset 0x00000000
Width 32

3.3.175  ram_ecc_errd_int_info_63_32
Shows information relating to the interrupt
The ram_ecc_errd_int_info_63_32 register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.
Configurations

There is only one DMC configuration.

Attributes

Offset 0x524
Type Read-only
Reset 0x00000000
Width 32

3.3.176 dram_ecc_errc_int_info_31_00

Shows information relating to the interrupt
The dram_ecc_errc_int_info_31_00 register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes

Offset 0x528
Type Read-only
Reset 0x00000000
Width 32

3.3.177 dram_ecc_errc_int_info_63_32

Shows information relating to the interrupt
The dram_ecc_errc_int_info_63_32 register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes

Offset 0x52C
Type Read-only
Reset 0x00000000
Width 32

3.3.178 dram_ecc_errd_int_info_31_00

Shows information relating to the interrupt
The dram_ecc_errd_int_info_31_00 register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes

Offset 0x530
Type Read-only
Reset 0x00000000
Width 32
3.3.179  **dram_ecc_errd_int_info_63_32**

Shows information relating to the interrupt

The dram_ecc_errd_int_info_63_32 register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be changed.

**Configurations**
There is only one DMC configuration.

**Attributes**
- **Offset**: 0x534
- **Type**: Read-only
- **Reset**: 0x00000000
- **Width**: 32

3.3.180  **failed_access_int_info_31_00**

Shows information relating to the interrupt

The failed_access_int_info_31_00 register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be changed.

**Configurations**
There is only one DMC configuration.

**Attributes**
- **Offset**: 0x538
- **Type**: Read-only
- **Reset**: 0x00000000
- **Width**: 32

3.3.181  **failed_access_int_info_63_32**

Shows information relating to the interrupt

The failed_access_int_info_63_32 register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be changed.

**Configurations**
There is only one DMC configuration.

**Attributes**
- **Offset**: 0x53C
- **Type**: Read-only
- **Reset**: 0x00000000
- **Width**: 32

3.3.182  **failed_prog_int_info_31_00**

Shows information relating to the interrupt

The failed_prog_int_info_31_00 register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be changed.

**Configurations**
There is only one DMC configuration.
Attributes

Offset  0x540
Type    Read-only
Reset   0x00000000
Width   32

3.3.183 failed_prog_int_info_63_32

Shows information relating to the interrupt

The failed_prog_int_info_63_32 register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes

Offset  0x544
Type    Read-only
Reset   0x00000000
Width   32

3.3.184 link_err_int_info_31_00

Shows information relating to the interrupt

The link_err_int_info_31_00 register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes

Offset  0x548
Type    Read-only
Reset   0x00000000
Width   32

3.3.185 link_err_int_info_63_32

Shows information relating to the interrupt

The link_err_int_info_63_32 register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes

Offset  0x54C
Type    Read-only
Reset   0x00000000
Width   32
3.3.186 arch_fsm_int_info_31_00

Shows information relating to the interrupt

The arch_fsm_int_info_31_00 register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x550</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-only</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.187 arch_fsm_int_info_63_32

Shows information relating to the interrupt

The arch_fsm_int_info_63_32 register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x554</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-only</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.188 integ_cfg

Integration test register to enable integration test mode.

The integ_cfg register characteristics are:

Usage constraints
Can be read from when in ALL states. Can be written to when in CONFIG or LOW-POWER states.

Configurations
There is only one DMC configuration.

Attributes

<table>
<thead>
<tr>
<th>Offset</th>
<th>0xE00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-write</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.189 integ_outputs

Drives the value of outputs when in integration test mode.

The integ_outputs register characteristics are:

Usage constraints
Cannot be read from. Can be written to when in CONFIG or LOW-POWER states.
Configurations
There is only one DMC configuration.

Attributes
Offset 0xE08
Type Write-only
Reset 0x00000000
Width 32

3.3.190 address_control_now
Configures the DRAM address parameters. Use the DRAM device data sheet or Serial Presence Detect (SPD)-derived values to assist in programming these values.

The address_control_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x1010
Type Read-only
Reset 0x00030202
Width 32

3.3.191 decode_control_now
Configures how the DRAM address is decoded from the system address. The DRAM address consists of the rank, bank, row address, and the column address.

The decode_control_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x1014
Type Read-only
Reset 0x00000000
Width 32

3.3.192 address_map_now
Configures the system address mapping options.

The address_map_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

Configurations
There is only one DMC configuration.

Attributes
3.3.193  low_power_control_now

Configures the low-power features of the DMC.

The low_power_control_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x101C
- Type: Read-only
- Reset: 0x00000000
- Width: 32

3.3.194  turnaround_control_now

Configures the settings for arbitration between read and write and rank to rank traffic on the DRAM bus.

The turnaround_control_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x1028
- Type: Read-only
- Reset: 0xF0F0F0F
- Width: 32

3.3.195  hit_turnaround_control_now

Configures the settings for preventing starvation of non-hits in the presence of in-row hit streams.

The hit_turnaround_control_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x102C
- Type: Read-only
- Reset: 0x8909FBF
- Width: 32
3.3.196 **qos_class_control_now**

Configures the priority class for each QoS encoding.

The qos_class_control_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x1030
- Type: Read-only
- Reset: 0x00000FC8
- Width: 32

3.3.197 **escalation_control_now**

Configures the settings for escalating the priority of entries in the queue.

The escalation_control_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x1034
- Type: Read-only
- Reset: 0x00080F03
- Width: 32

3.3.198 **qv_control_31_00_now**

Configures the priority settings for each QoS encoding.

The qv_control_31_00_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x1038
- Type: Read-only
- Reset: 0x76543210
- Width: 32

3.3.199 **qv_control_63_32_now**

Configures the priority settings for each QoS encoding.

The qv_control_63_32_now register characteristics are:
Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x103C
Type Read-only
Reset 0xFEDCBA98
Width 32

3.3.200 rt_control_31_00_now
Configures the timeout settings for each QoS encoding.
The rt_control_31_00_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x1040
Type Read-only
Reset 0x00000000
Width 32

3.3.201 rt_control_63_32_now
Configures the timeout settings for each QoS encoding.
The rt_control_63_32_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x1044
Type Read-only
Reset 0x00000000
Width 32

3.3.202 timeout_control_now
Configures the prescaler applied to timeout values.
The timeout_control_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
3.3.203 credit_control_now

Configures the settings for preventing starvation of CHI protocol retries.

The credit_control_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x104C
- Type: Read-only
- Reset: 0x00000F03
- Width: 32

3.3.204 write_priority_control_31_00_now

Configures the priority settings for write requests within the DMC.

The write_priority_control_31_00_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x1050
- Type: Read-only
- Reset: 0x00000000
- Width: 32

3.3.205 write_priority_control_63_32_now

Configures the priority settings for write requests within the DMC.

The write_priority_control_63_32_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x1054
- Type: Read-only
- Reset: 0x00000000
- Width: 32
3.3.206 queue_threshold_control_31_00_now

Configures the threshold settings for requests in the DMC

The queue_threshold_control_31_00_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset  0x1060
Type    Read-only
Reset   0x00000000
Width   32

3.3.207 queue_threshold_control_63_32_now

Configures the threshold settings for requests in the DMC

The queue_threshold_control_63_32_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset  0x1064
Type    Read-only
Reset   0x00000000
Width   32

3.3.208 memory_address_max_31_00_now

Configures the address space control for the DMC default region.

The memory_address_max_31_00_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset  0x1078
Type    Read-only
Reset   0x00000010
Width   32

3.3.209 memory_address_max_43_32_now

Configures the address space control for the DMC default region.

The memory_address_max_43_32_now register characteristics are:
Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG,
LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x107C
Type Read-only
Reset 0x00000000
Width 32

3.3.210  access_address_min0_31_00_now
Configures the address space control for address region 0.
The access_address_min0_31_00_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG,
LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x1080
Type Read-only
Reset 0x00000000
Width 32

3.3.211  access_address_min0_43_32_now
Configures the address space control for address region 0.
The access_address_min0_43_32_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG,
LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x1084
Type Read-only
Reset 0x00000000
Width 32

3.3.212  access_address_max0_31_00_now
Configures the address space control for address region 0.
The access_address_max0_31_00_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG,
LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
3.3.213 access_address_max0_43_32_now

Configures the address space control for address region 0.

The access_address_max0_43_32_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x1088</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-only</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.214 access_address_min1_31_00_now

Configures the address space control for address region 1.

The access_address_min1_31_00_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x1090</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-only</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.215 access_address_min1_43_32_now

Configures the address space control for address region 1.

The access_address_min1_43_32_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x1094</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-only</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>
### access_address_max1_31_00_now

Configures the address space control for address region 1.

The `access_address_max1_31_00_now` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
- There is only one DMC configuration.

**Attributes**
- **Offset**: 0x1098
- **Type**: Read-only
- **Reset**: 0x00000000
- **Width**: 32

### access_address_max1_43_32_now

Configures the address space control for address region 1.

The `access_address_max1_43_32_now` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
- There is only one DMC configuration.

**Attributes**
- **Offset**: 0x109C
- **Type**: Read-only
- **Reset**: 0x00000000
- **Width**: 32

### access_address_min2_31_00_now

Configures the address space control for address region 2.

The `access_address_min2_31_00_now` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
- There is only one DMC configuration.

**Attributes**
- **Offset**: 0x10A0
- **Type**: Read-only
- **Reset**: 0x00000000
- **Width**: 32

### access_address_min2_43_32_now

Configures the address space control for address region 2.

The `access_address_min2_43_32_now` register characteristics are:
Usage constraints
  Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
  There is only one DMC configuration.

Attributes
  Offset  0x10A4
  Type    Read-only
  Reset   0x00000000
  Width   32

3.3.220  access_address_max2_31_00_now

Configures the address space control for address region 2.

The access_address_max2_31_00_now register characteristics are:

Usage constraints
  Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
  There is only one DMC configuration.

Attributes
  Offset  0x10A8
  Type    Read-only
  Reset   0x00000000
  Width   32

3.3.221  access_address_max2_43_32_now

Configures the address space control for address region 2.

The access_address_max2_43_32_now register characteristics are:

Usage constraints
  Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
  There is only one DMC configuration.

Attributes
  Offset  0x10AC
  Type    Read-only
  Reset   0x00000000
  Width   32

3.3.222  access_address_min3_31_00_now

Configures the address space control for address region 3.

The access_address_min3_31_00_now register characteristics are:

Usage constraints
  Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
  There is only one DMC configuration.

Attributes
3.3.223 **access_address_min3_43_32_now**

Configures the address space control for address region 3.

The `access_address_min3_43_32_now` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
- There is only one DMC configuration.

**Attributes**
- **Offset**: 0x10B0
- **Type**: Read-only
- **Reset**: 0x00000000
- **Width**: 32

3.3.224 **access_address_max3_31_00_now**

Configures the address space control for address region 3.

The `access_address_max3_31_00_now` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
- There is only one DMC configuration.

**Attributes**
- **Offset**: 0x10B4
- **Type**: Read-only
- **Reset**: 0x00000000
- **Width**: 32

3.3.225 **access_address_max3_43_32_now**

Configures the address space control for address region 3.

The `access_address_max3_43_32_now` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
- There is only one DMC configuration.

**Attributes**
- **Offset**: 0x10B8
- **Type**: Read-only
- **Reset**: 0x00000000
- **Width**: 32
3.3.226  access_address_min4_31_00_now

Configures the address space control for address region 4.

The access_address_min4_31_00_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x10C0
Type Read-only
Reset 0x00000000
Width 32

3.3.227  access_address_min4_43_32_now

Configures the address space control for address region 4.

The access_address_min4_43_32_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x10C4
Type Read-only
Reset 0x00000000
Width 32

3.3.228  access_address_max4_31_00_now

Configures the address space control for address region 4.

The access_address_max4_31_00_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x10C8
Type Read-only
Reset 0x00000000
Width 32

3.3.229  access_address_max4_43_32_now

Configures the address space control for address region 4.

The access_address_max4_43_32_now register characteristics are:
Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x10CC
Type Read-only
Reset 0x00000000
Width 32

3.3.230 access_address_min5_31_00_now
Configures the address space control for address region 5.

The access_address_min5_31_00_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x10D0
Type Read-only
Reset 0x00000000
Width 32

3.3.231 access_address_min5_43_32_now
Configures the address space control for address region 5.

The access_address_min5_43_32_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x10D4
Type Read-only
Reset 0x00000000
Width 32

3.3.232 access_address_max5_31_00_now
Configures the address space control for address region 5.

The access_address_max5_31_00_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
3.3.233  access_address_max5_43_32_now

Configures the address space control for address region 5.

The access_address_max5_43_32_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
  Offset 0x10D8
  Type Read-only
  Reset 0x00000000
  Width 32

3.3.234  access_address_min6_31_00_now

Configures the address space control for address region 6.

The access_address_min6_31_00_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
  Offset 0x10DC
  Type Read-only
  Reset 0x00000000
  Width 32

3.3.235  access_address_min6_43_32_now

Configures the address space control for address region 6.

The access_address_min6_43_32_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
  Offset 0x10E0
  Type Read-only
  Reset 0x00000000
  Width 32
3.3.236 **access_address_max6_31_00_now**

Configures the address space control for address region 6.

The access_address_max6_31_00_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x10E8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-only</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.237 **access_address_max6_43_32_now**

Configures the address space control for address region 6.

The access_address_max6_43_32_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x10EC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-only</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.238 **access_address_min7_31_00_now**

Configures the address space control for address region 7.

The access_address_min7_31_00_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x10F0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-only</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.239 **access_address_min7_43_32_now**

Configures the address space control for address region 7.

The access_address_min7_43_32_now register characteristics are:
Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x10F4
Type Read-only
Reset 0x00000000
Width 32

3.3.240 access_address_max7_31_00_now

Configures the address space control for address region 7.

The access_address_max7_31_00_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x10F8
Type Read-only
Reset 0x00000000
Width 32

3.3.241 access_address_max7_43_32_now

Configures the address space control for the address region 7.

The access_address_max7_43_32_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x10FC
Type Read-only
Reset 0x00000000
Width 32

3.3.242 dci_replay_type_now

Configures the behavior of the DMC if a DRAM or PHY error is received when executing a direct command.

The dci_replay_type_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, PAUSED or READY states.

Configurations
There is only one DMC configuration.
Attributes
Offset 0x1110
Type Read-only
Reset 0x00000002
Width 32

3.3.243 refresh_control_now
Configures the type of refresh commands issued by the DMC.
The refresh_control_now register characteristics are:
Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or PAUSED states.
Configurations
There is only one DMC configuration.
Attributes
Offset 0x1120
Type Read-only
Reset 0x00000000
Width 32

3.3.244 memory_type_now
Configures the DMC for the attached memory type.
The memory_type_now register characteristics are:
Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.
Configurations
There is only one DMC configuration.
Attributes
Offset 0x1128
Type Read-only
Reset 0x00000101
Width 32

3.3.245 scrub_control0_now
Scrub engine channel control register.
The scrub_control0_now register characteristics are:
Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.
Configurations
There is only one DMC configuration.
Attributes
Offset 0x1170
Type Read-only
Reset 0x1F000000
3.3.246 **scrub_address_min0_now**

Configures the address space control for the scrub engine channel.

The `scrub_address_min0_now` register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x1174
- Type: Read-only
- Reset: 0x00000000
- Width: 32

3.3.247 **scrub_address_max0_now**

Configures the address space control for the scrub engine channel.

The `scrub_address_max0_now` register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x1178
- Type: Read-only
- Reset: 0x00000000
- Width: 32

3.3.248 **scrub_control1_now**

Scrub engine channel control register.

The `scrub_control1_now` register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x1180
- Type: Read-only
- Reset: 0x1F000000
- Width: 32

3.3.249 **scrub_address_min1_now**

Configures the address space control for the scrub engine channel.

The `scrub_address_min1_now` register characteristics are:
Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x1184
Type Read-only
Reset 0x00000000
Width 32

3.3.250 scrub_address_max1_now

Configures the address space control for the scrub engine channel.

The scrub_address_max1_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x1188
Type Read-only
Reset 0x00000000
Width 32

3.3.251 scrub_control2_now

Scrub engine channel control register.

The scrub_control2_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x1190
Type Read-only
Reset 0x1F000000
Width 32

3.3.252 scrub_address_min2_now

Configures the address space control for the scrub engine channel.

The scrub_address_min2_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

Configurations
There is only one DMC configuration.

Attributes
### 3.3.253 scrub_address_max2_now

Configures the address space control for the scrub engine channel.

The `scrub_address_max2_now` register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x1198</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-only</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

### 3.3.254 scrub_control3_now

Scrub engine channel control register.

The `scrub_control3_now` register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x11A0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-only</td>
</tr>
<tr>
<td>Reset</td>
<td>0x1F000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

### 3.3.255 scrub_address_min3_now

Configures the address space control for the scrub engine channel.

The `scrub_address_min3_now` register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x11A4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-only</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>
3.3.256  **scrub_address_max3_now**

Configures the address space control for the scrub engine channel.

The `scrub_address_max3_now` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

**Configurations**
- There is only one DMC configuration.

**Attributes**
- **Offset**: 0x11A8
- **Type**: Read-only
- **Reset**: 0x00000000
- **Width**: 32

3.3.257  **scrub_control4_now**

Scrub engine channel control register.

The `scrub_control4_now` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

**Configurations**
- There is only one DMC configuration.

**Attributes**
- **Offset**: 0x11B0
- **Type**: Read-only
- **Reset**: 0x1F000000
- **Width**: 32

3.3.258  **scrub_address_min4_now**

Configures the address space control for the scrub engine channel.

The `scrub_address_min4_now` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

**Configurations**
- There is only one DMC configuration.

**Attributes**
- **Offset**: 0x11B4
- **Type**: Read-only
- **Reset**: 0x00000000
- **Width**: 32

3.3.259  **scrub_address_max4_now**

Configures the address space control for the scrub engine channel.

The `scrub_address_max4_now` register characteristics are:
Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x11B8
Type Read-only
Reset 0x00000000
Width 32

3.3.260 scrub_control5_now
Scrub engine channel control register.

The scrub_control5_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x11C0
Type Read-only
Reset 0x1F000000
Width 32

3.3.261 scrub_address_min5_now
Configures the address space control for the scrub engine channel.

The scrub_address_min5_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x11C4
Type Read-only
Reset 0x00000000
Width 32

3.3.262 scrub_address_max5_now
Configures the address space control for the scrub engine channel.

The scrub_address_max5_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

Configurations
There is only one DMC configuration.

Attributes

Non-Confidential
3.3.263 scrub_control6_now
Scrub engine channel control register.
The scrub_control6_now register characteristics are:
Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.
Configurations
There is only one DMC configuration.
Attributes
  Offset 0x11D0
  Type  Read-only
  Reset 0x1F000000
  Width 32

3.3.264 scrub_address_min6_now
Configures the address space control for the scrub engine channel.
The scrub_address_min6_now register characteristics are:
Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.
Configurations
There is only one DMC configuration.
Attributes
  Offset 0x11D4
  Type  Read-only
  Reset 0x00000000
  Width 32

3.3.265 scrub_address_max6_now
Configures the address space control for the scrub engine channel.
The scrub_address_max6_now register characteristics are:
Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.
Configurations
There is only one DMC configuration.
Attributes
  Offset 0x11D8
  Type  Read-only
  Reset 0x00000000
  Width 32
3.3.266  `scrub_control7_now`
Scrub engine channel control register.

The `scrub_control7_now` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

**Configurations**
- There is only one DMC configuration.

**Attributes**
- **Offset**: 0x11E0
- **Type**: Read-only
- **Reset**: 0x1F000000
- **Width**: 32

3.3.267  `scrub_address_min7_now`
Configures the address space control for the scrub engine channel.

The `scrub_address_min7_now` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

**Configurations**
- There is only one DMC configuration.

**Attributes**
- **Offset**: 0x11E4
- **Type**: Read-only
- **Reset**: 0x00000000
- **Width**: 32

3.3.268  `scrub_address_max7_now`
Configures the address space control for the scrub engine channel.

The `scrub_address_max7_now` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

**Configurations**
- There is only one DMC configuration.

**Attributes**
- **Offset**: 0x11E8
- **Type**: Read-only
- **Reset**: 0x00000000
- **Width**: 32

3.3.269  `feature_control_now`
Control register for DMC features.

The `feature_control_now` register characteristics are:
3.3.270 mux_control_now

Control muxing options for the DMC.

The mux_control_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x11F0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-only</td>
</tr>
<tr>
<td>Reset</td>
<td>0x0AA000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.271 rank_remap_control_now

Control register for rank remap.

The rank_remap_control_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x11F4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-only</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.272 scrub_control_now

Scrub engine channel control register.

The scrub_control_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG or LOW-POWER states.

**Configurations**
There is only one DMC configuration.

**Attributes**
3.3.273  **t_refi_now**

Configures the refresh interval timing parameter. It must be programmed to the device average all-bank AUTOREFRESH interval, divided by 8.

The **t_refi_now** register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x11FC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-only</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00001F00</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.274  **t_rfc_now**

Configures the tRFC timing parameter. This determines the delay applied after an AUTOREFRESH command before any other command is issued to the same rank.

The **t_rfc_now** register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x1200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-only</td>
</tr>
<tr>
<td>Reset</td>
<td>0x000090100</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.275  **t_mrr_now**

Configures the tMRR timing parameter. This determines the Mode Register Read (including Multi-Purpose Register Reads) command delay before any other command is issued to the same rank.

The **t_mrr_now** register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x1204</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-only</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00008C23</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>
3.3.276  t_mrw_now

Configures the tMRW timing parameter. This determines the delay applied after a Mode Register Write (including Multi-Purpose Register Writes) command before any other command is issued to the same rank.

The t_mrw_now register characteristics are:

Usage constraints
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
- There is only one DMC configuration.

Attributes
- Offset: 0x120C
- Type: Read-only
- Reset: 0x00000002
- Width: 32

3.3.277  t_rdpden_now

Configures the tRDPDEN timing parameter. This determines the delay applied after a Read command before a power down command can be issued to the same rank.

The t_rdpden_now register characteristics are:

Usage constraints
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
- There is only one DMC configuration.

Attributes
- Offset: 0x1210
- Type: Read-only
- Reset: 0x0000000A
- Width: 32

3.3.278  t_rcd_now

Configures the tRCD timing parameter. This determines the delay applied after an ACTIVATE command before a READ or WRITE command is issued to the same bank.

The t_rcd_now register characteristics are:

Usage constraints
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
- There is only one DMC configuration.

Attributes
- Offset: 0x1218
- Type: Read-only
- Reset: 0x00000005
- Width: 32
3.3.279 t_ras_now

Configures the tRAS timing parameter. This determines the delay applied after an ACTIVATE command before a PRECHARGE command is issued to the same bank.

The t_ras_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
- Offset: 0x121C
- Type: Read-only
- Reset: 0x0000000E
- Width: 32

3.3.280 t_rp_now

Configures the tRP timing parameter. This determines the delay applied after a PRECHARGE command before any other command is issued to the same bank.

The t_rp_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
- Offset: 0x1220
- Type: Read-only
- Reset: 0x00000005
- Width: 32

3.3.281 t_rpall_now

Configures the tRPALL timing parameter. This determines the delay applied after a PRECHARGEALL command before any other command is issued to the same rank.

The t_rpall_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
- Offset: 0x1224
- Type: Read-only
- Reset: 0x00000005
- Width: 32
3.3.282 **t_rrd_now**

Configures the tRRD timing parameter. This determines the delay applied after an ACTIVATE command before another ACTIVATE command is issued to the same rank. The _l and _s fields apply to the same bank group, and a different bank group, respectively, as described in the DDR4 specification.

The `t_rrd_now` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
- There is only one DMC configuration.

**Attributes**

- **Offset**: 0x1228
- **Type**: Read-only
- **Reset**: 0x00000404
- **Width**: 32

3.3.283 **t_act_window_now**

Configures the tFAW and tMAWi timing parameters.

The `t_act_window_now` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
- There is only one DMC configuration.

**Attributes**

- **Offset**: 0x122C
- **Type**: Read-only
- **Reset**: 0x03560014
- **Width**: 32

3.3.284 **t_rtr_now**

Configures the read-to-read timing parameter. This determines the READ to READ command delay applied between reads to the same chip, other bank group (`t_rtr_s`), same chip, same bank group (`t_rtr_l`), and different chip-selects (`t_rtr_cs`).

The `t_rtr_now` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
- There is only one DMC configuration.

**Attributes**

- **Offset**: 0x1234
- **Type**: Read-only
- **Reset**: 0x00060404
- **Width**: 32
3.3.285 \texttt{t\_rtw\_now}

Configures the read-to-write timing parameter. This determines the READ to WRITE command delay applied between issued commands to the same chip, other bank group (t\_rtw\_s), same chip, same bank group (t\_trw\_l), and other chip-selects (t\_rtw\_cs).

The \texttt{t\_rtw\_now} register characteristics are:

\textbf{Usage constraints}
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

\textbf{Configurations}
There is only one DMC configuration.

\textbf{Attributes}
\begin{itemize}
  \item Offset \(0x1238\)
  \item Type \textit{Read-only}
  \item Reset \(0x00060606\)
  \item Width 32
\end{itemize}

3.3.286 \texttt{t\_rtp\_now}

Configures the read-to-precharge timing parameter. This determines the READ to PRECHARGE command delay applied between issued commands to the same bank.

The \texttt{t\_rtp\_now} register characteristics are:

\textbf{Usage constraints}
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

\textbf{Configurations}
There is only one DMC configuration.

\textbf{Attributes}
\begin{itemize}
  \item Offset \(0x123C\)
  \item Type \textit{Read-only}
  \item Reset \(0x00000004\)
  \item Width 32
\end{itemize}

3.3.287 \texttt{t\_wr\_now}

Configures the tWR timing parameter. This determines the write recovery time and is used as the delay applied between the issue of a WRITE command and subsequent commands, other than WRITEs, to the same bank.

The \texttt{t\_wr\_now} register characteristics are:

\textbf{Usage constraints}
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

\textbf{Configurations}
There is only one DMC configuration.

\textbf{Attributes}
\begin{itemize}
  \item Offset \(0x1244\)
  \item Type \textit{Read-only}
  \item Reset \(0x00000005\)
  \item Width 32
\end{itemize}
3.3.288  **t_wtr_now**

Configures the write-to-read timing parameter, for both same chip, other bank group (tWTR_s), same chip, same bank group (t_WTR_l), and alternate chip (tWTR_cs).

The **t_wtr_now** register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- **Offset**: 0x1248
- **Type**: Read-only
- **Reset**: 0x00040505
- **Width**: 32

3.3.289  **t_wtw_now**

Configures the write-to-write timing parameter for same chip, other bank group (t_wtw_s), same chip, same bank group (t_wtw_l), alternate chip (t_wtw_cs) writes.

The **t_wtw_now** register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- **Offset**: 0x124C
- **Type**: Read-only
- **Reset**: 0x00060404
- **Width**: 32

3.3.290  **t_xmpd_now**

Configures the command delay between exiting Maximum Power Down and a subsequent command to that rank.

The **t_xmpd_now** register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- **Offset**: 0x1254
- **Type**: Read-only
- **Reset**: 0x000003FF
- **Width**: 32
3.3.291 t_ep_now

Configures the enter power-down timing parameter. This parameter is applied between the issue of an active or precharge power down request and subsequent commands to the same rank.

The t_ep_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
- Offset: 0x1258
- Type: Read-only
- Reset: 0x00000002
- Width: 32

3.3.292 t_xp_now

Configures the exit power-down timing parameter for operations that do not require a DLL (tXP), and those that do (tXPDLL).

The t_xp_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
- Offset: 0x125C
- Type: Read-only
- Reset: 0x00060002
- Width: 32

3.3.293 t_esr_now

Configures the enter self-refresh timing parameter. This parameter is applied between issue of an enter self-refresh request and subsequent commands to the same rank.

The t_esr_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG,LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
- Offset: 0x1260
- Type: Read-only
- Reset: 0x0000000E
- Width: 32
3.3.294  **t_xsr_now**

Configures the exit self-refresh timing parameter. This parameter is applied between the issue of an exit self-refresh request and subsequent commands to the same rank.

The t_xsr_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- **Offset**: 0x1264
- **Type**: Read-only
- **Reset**: 0x05120100
- **Width**: 32

3.3.295  **t_esrck_now**

Configures the delay between entering self-refresh and disabling the DRAM clock. This parameter is applied when stopping the clock when in self-refresh and when in a maximum power-down state.

The t_esrck_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- **Offset**: 0x1268
- **Type**: Read-only
- **Reset**: 0x00000005
- **Width**: 32

3.3.296  **t_ckxsr_now**

Configures the delay between DRAM clock enable and exiting self-refresh. This parameter is applied when re-instating the clock when in self-refresh and when in a maximum power-down state.

The t_ckxsr_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- **Offset**: 0x126C
- **Type**: Read-only
- **Reset**: 0x00000001
- **Width**: 32

3.3.297  **t_cmd_now**

Configures command signaling timing.
The t_cmd_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x1270
- Type: Read-only
- Reset: 0x00000000
- Width: 32

### 3.3.298 t_parity_now

Parity latencies t_parinlat and t_completion.

The t_parity_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x1274
- Type: Read-only
- Reset: 0x0000900
- Width: 32

### 3.3.299 t_zqcs_now

Configures the delay to apply following a ZQC-Short calibration command.

The t_zqcs_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x1278
- Type: Read-only
- Reset: 0x0000040
- Width: 32

### 3.3.300 t_rddata_en_now

Determines the time between a READ command commencing on the DFI interface, and the assertion of the dfi_read_en signal.

The t_rddata_en_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.
3.3.301 t_phyrdlat_now

Determines the maximum possible time between the assertion of the dfi_read_en signal, and the assertion of the dfi_rddata_valid signal by the PHY.

The t_phyrdlat_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x1300
Type Read-only
Reset 0x00000001
Width 32

3.3.302 t_phywrlat_now

Determines the time between a WRITE command commencing on the DFI interface, and the assertion of the dfi_wrdata_en, dfi_wrdata_cs and dfi_wrdata signals.

The t_phywrlat_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset 0x1304
Type Read-only
Reset 0x00000000
Width 32

3.3.303 rdlvl_control_now

Determines the DMC behavior during read training operations. See the PHY training interface section of the Integration Manual for more details on PHY training.

The rdlvl_control_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
3.3.304  rdlvl_mrs_now

Determines the Mode Register command to use to place the DRAM into a training mode for read training, when enabled by the rdlvl_control register. See the PHY interface section of the Integration Manual for more information on PHY training.

The rdlvl_mrs_now register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
- There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x1310</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-only</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00001080</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.305  t_rdlvl_en_now

Configures the t_rdlvl_en timing parameter. This specifies the cycle delay between asserting dfi_rdlvl_en and the first training command, and also the cycle delay between deasserting dfi_rdlvl_en and performing any subsequent command. It also specifies the minimum delay between training commands and refreshes during training.

The t_rdlvl_en_now register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
- There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x1314</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-only</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000004</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.306  t_rdlvl_rr_now

Configures the t_rdlvl_rr timing parameter. This specifies the cycle delay between training commands. It also specifies the minimum delay between the last training command and deasserting dfi_rdlvl_en after observing dfi_rdlvl_resp.

The t_rdlvl_rr_now register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
- There is only one DMC configuration.
Attributes
Offset  0x131C
Type    Read-only
Reset   0x00000000
Width   32

3.3.307  wrlvl_control_now

Determines the DMC behavior during write training operations. See the PHY training interface section of the Integration Manual for more information on PHY training.

The wrlvl_control_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset  0x1320
Type    Read-only
Reset   0x00001000
Width   32

3.3.308  wrlvl_mrs_now

Determines the Mode Register command that the DMC must use to put the DRAM into a training mode for write leveling. You enable this function with the wrlvl_control Register. See the PHY training interface section of the Integration Manual for more information.

The wrlvl_mrs_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

Configurations
There is only one DMC configuration.

Attributes
Offset  0x1324
Type    Read-only
Reset   0x00000086
Width   32

3.3.309  t_wrlvl_en_now

Configures the t_wrlvl_en timing parameter. Specifies the cycle delay between asserting ODT for training and asserting dfi_wrlvl_en, the delay between asserting dfi_wrlvl_en and the first training command, the delay between deasserting dfi_wrlvl_en and de-asserting ODT, and deasserting ODT to any subsequent command. It is also used between ODT transitions and refreshes generated during training.

The t_wrlvl_en_now register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.
3.3.310  **t_wrlvl_ww_now**

Configures the \texttt{t\_wrlvl\_ww} timing parameter. Specifies the cycle delay between training commands. Also specifies the minimum delay between the last training command and de-asserting \texttt{dfi\_wrlvl\_en} on observing \texttt{dfi\_wrlvl\_resp}.

The \texttt{t_wrlvl_ww_now} register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in \texttt{CONFIG}, \texttt{LOW-POWER} or \texttt{PAUSED} states.

**Configurations**
- There is only one DMC configuration.

**Attributes**
- Offset 0x1328
- Type Read-only
- Reset 0x00000000
- Width 32

3.3.311  **phy_power_control_now**

Configures the low-power requests made to the PHY for the different channel states.

The \texttt{phy\_power\_control\_now} register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in \texttt{CONFIG}, \texttt{LOW-POWER} or \texttt{PAUSED} states.

**Configurations**
- There is only one DMC configuration.

**Attributes**
- Offset 0x132C
- Type Read-only
- Reset 0x00000000
- Width 32

3.3.312  **t_lpresp_now**

Configures the minimum cycle delay to apply for PHY low-power handshakes.

The \texttt{t\_lpresp\_now} register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in \texttt{CONFIG}, \texttt{LOW-POWER} or \texttt{PAUSED} states.

**Configurations**
- There is only one DMC configuration.

**Attributes**
- Offset 0x1348
- Type Read-only
- Reset 0x00000000
- Width 32
3.3.313 phy_update_control_now

Configures the update mechanism to use in response to PHY training requests.

The phy_update_control_now register characteristics are:

- **Usage constraints**: Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.
- **Configurations**: There is only one DMC configuration.
- **Attributes**
  - Offset: 0x134C
  - Type: Read-only
  - Reset: 0x00000000
  - Width: 32

3.3.314 odt_timing_now

Configures the ODT on and off timing.

The odt_timing_now register characteristics are:

- **Usage constraints**: Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.
- **Configurations**: There is only one DMC configuration.
- **Attributes**
  - Offset: 0x1350
  - Type: Read-only
  - Reset: 0x0FE00000
  - Width: 32

3.3.315 odt_wr_control_31_00_now

Configures the ODT on and off settings for active and inactive ranks during writes.

The odt_wr_control_31_00_now register characteristics are:

- **Usage constraints**: Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.
- **Configurations**: There is only one DMC configuration.
- **Attributes**
  - Offset: 0x1360
  - Type: Read-only
  - Reset: 0x08040201
  - Width: 32
3.3.316  **odt_wr_control_63_32_now**

Configures the ODT on and off settings for active and inactive ranks during writes.

The `odt_wr_control_63_32_now` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
- There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Reset</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x1364</td>
<td>Read-only</td>
<td>0x80402010</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.317  **odt_rd_control_31_00_now**

Configures the ODT on and off settings for active and inactive ranks during reads.

The `odt_rd_control_31_00_now` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
- There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Reset</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x1368</td>
<td>Read-only</td>
<td>0x00000000</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.318  **odt_rd_control_63_32_now**

Configures the ODT on and off settings for active and inactive ranks during reads.

The `odt_rd_control_63_32_now` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
- There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Reset</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x136C</td>
<td>Read-only</td>
<td>0x00000000</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.319  **dq_map_control_15_00_now**

Controls the DQ mapping compensation applied for CRC calculation. For each nibble of the DQ bus, the DIMM SPD defines a DQ Map Index to define the bit connectivity to each rank. Program the DQ Map Index retrieved from the SPD for each nibble into the corresponding register in the DMC for correct CRC operation.
The dq_map_control_15_00_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x1380</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-only</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.320 **dq_map_control_31_16_now**

Controls the DQ mapping compensation applied for CRC calculation. For each nibble of the DQ bus, the DIMM SPD defines a DQ Map Index to define the bit connectivity to each rank. Program the DQ Map Index retrieved from the SPD for each nibble into the corresponding register in the DMC for correct CRC operation.

The dq_map_control_31_16_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x1384</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-only</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>

3.3.321 **dq_map_control_47_32_now**

Controls the DQ mapping compensation applied for CRC calculation. For each nibble of the DQ bus, the DIMM SPD defines a DQ Map Index to define the bit connectivity to each rank. Program the DQ Map Index retrieved from the SPD for each nibble into the corresponding register in the DMC for correct CRC operation.

The dq_map_control_47_32_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
There is only one DMC configuration.

**Attributes**

<table>
<thead>
<tr>
<th>Offset</th>
<th>0x1388</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Read-only</td>
</tr>
<tr>
<td>Reset</td>
<td>0x00000000</td>
</tr>
<tr>
<td>Width</td>
<td>32</td>
</tr>
</tbody>
</table>
3.3.322  **dq_map_control_63_48_now**

Controls the DQ mapping compensation applied for CRC calculation. For each nibble of the DQ bus, the DIMM SPD defines a DQ Map Index to define the bit connectivity to each rank. Program the DQ Map Index retrieved from the SPD for each nibble into the corresponding register in the DMC for correct CRC operation.

The `dq_map_control_63_48_now` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
- There is only one DMC configuration.

**Attributes**
- Offset: 0x138C
- Type: Read-only
- Reset: 0x00000000
- Width: 32

3.3.323  **dq_map_control_71_64_now**

Controls the DQ mapping compensation applied for CRC calculation. For each nibble of the DQ bus, the DIMM SPD defines a DQ Map Index to define the bit connectivity to each rank. Program the DQ Map Index retrieved from the SPD for DIMM Check Bits bus into this register in the DMC for correct CRC operation.

The `dq_map_control_71_64_now` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in CONFIG, LOW-POWER or PAUSED states.

**Configurations**
- There is only one DMC configuration.

**Attributes**
- Offset: 0x1390
- Type: Read-only
- Reset: 0x00000000
- Width: 32

3.3.324  **user_config0_now**

Drives the output `user_config0` signal.

The `user_config0_now` register characteristics are:

**Usage constraints**
- Can be read from when in ALL states. Cannot be written to and only updated when in ALL states.

**Configurations**
- There is only one DMC configuration.

**Attributes**
- Offset: 0x1408
- Type: Read-only
- Reset: 0x00000000
- Width: 32
3.3.325  **user_config1_now**

Drives the output user_config1 signal.

The user_config1_now register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be written to and only updated when in ALL states.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x140C
- Type: Read-only
- Reset: 0x00000000
- Width: 32

3.3.326  **periph_id_4**

Peripheral ID register.

The periph_id_4 register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be changed.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x1FD0
- Type: Read-only
- Reset: 0x00000014
- Width: 32

3.3.327  **periph_id_0**

Peripheral ID register.

The periph_id_0 register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be changed.

**Configurations**
There is only one DMC configuration.

**Attributes**
- Offset: 0x1FE0
- Type: Read-only
- Reset: 0x00000052
- Width: 32

3.3.328  **periph_id_1**

Peripheral ID register.

The periph_id_1 register characteristics are:

**Usage constraints**
Can be read from when in ALL states. Cannot be changed.
There is only one DMC configuration.

Attributes

- Offset: 0x1FE4
- Type: Read-only
- Reset: 0x000000B4
- Width: 32

3.3.329 periph_id_2

Peripheral ID register.

The periph_id_2 register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes

- Offset: 0x1FE8
- Type: Read-only
- Reset: 0x0000003B
- Width: 32

3.3.330 periph_id_3

Peripheral ID register.

The periph_id_3 register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes

- Offset: 0x1FEC
- Type: Read-only
- Reset: 0x00000000
- Width: 32

3.3.331 component_id_0

Component ID register.

The component_id_0 register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes

- Offset: 0x1FF0
- Type: Read-only
- Reset: 0x0000000D
- Width: 32
3.3.332 component_id_1

Component ID register.

The component_id_1 register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes
- Offset 0x1FF4
- Type Read-only
- Reset 0x000000F0
- Width 32

3.3.333 component_id_2

Component ID register.

The component_id_2 register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes
- Offset 0x1FF8
- Type Read-only
- Reset 0x00000005
- Width 32

3.3.334 component_id_3

Component ID register.

The component_id_3 register characteristics are:

Usage constraints
Can be read from when in ALL states. Cannot be changed.

Configurations
There is only one DMC configuration.

Attributes
- Offset 0x1FFC
- Type Read-only
- Reset 0x000000B1
- Width 32
Appendix A
Signal Descriptions

This appendix describes the DMC-520 signals.

It contains the following sections:
• \textit{A.1 Signals list} on page Appx-A-146.
A.1 Signals list

DMC signals list that excludes bus interface signals. The bus interface signals are defined by their own bus protocol standard.

The following table shows the Primary clock and reset signals list of the DMC.

**Table A-1 DMC Primary clock and reset signals list**

<table>
<thead>
<tr>
<th>Signal</th>
<th>Type</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clk</td>
<td>Input</td>
<td>1</td>
<td>Primary DMC clock</td>
</tr>
<tr>
<td>dfi_clk</td>
<td>Input</td>
<td>1</td>
<td>DFI clock</td>
</tr>
<tr>
<td>resetn</td>
<td>Input</td>
<td>1</td>
<td>Primary DMC reset</td>
</tr>
</tbody>
</table>

The following table shows the APB clock and reset signals list of the DMC.

**Table A-2 DMC APB clock and reset signals list**

<table>
<thead>
<tr>
<th>Signal</th>
<th>Type</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pclk</td>
<td>Input</td>
<td>1</td>
<td>APB clock</td>
</tr>
<tr>
<td>presetn</td>
<td>Input</td>
<td>1</td>
<td>APB reset</td>
</tr>
</tbody>
</table>

The following table shows the User I/O with APB access list of the DMC.

**Table A-3 DMC User I/O with APB access list**

<table>
<thead>
<tr>
<th>Signal</th>
<th>Type</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_status</td>
<td>Input</td>
<td>32</td>
<td>User defined inputs</td>
</tr>
<tr>
<td>user_config0</td>
<td>Output</td>
<td>32</td>
<td>User defined outputs</td>
</tr>
<tr>
<td>user_config1</td>
<td>Output</td>
<td>32</td>
<td>User defined outputs</td>
</tr>
<tr>
<td>user_config2</td>
<td>Output</td>
<td>32</td>
<td>User defined outputs</td>
</tr>
<tr>
<td>user_config3</td>
<td>Output</td>
<td>32</td>
<td>User defined outputs</td>
</tr>
<tr>
<td>user_periph_id_3</td>
<td>Input</td>
<td>8</td>
<td>Tie-off value to set the value of CMOD in the periph_id_3 bitfield</td>
</tr>
</tbody>
</table>

The following table shows the Events list of the DMC.

**Table A-4 DMC Events list**

<table>
<thead>
<tr>
<th>Signal</th>
<th>Type</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>scrub_event_in0</td>
<td>Input</td>
<td>1</td>
<td>Scrub event 0 trigger</td>
</tr>
<tr>
<td>scrub_event_in1</td>
<td>Input</td>
<td>1</td>
<td>Scrub event 1 trigger</td>
</tr>
<tr>
<td>scrub_event_in2</td>
<td>Input</td>
<td>1</td>
<td>Scrub event 2 trigger</td>
</tr>
<tr>
<td>scrub_event_in3</td>
<td>Input</td>
<td>1</td>
<td>Scrub event 3 trigger</td>
</tr>
<tr>
<td>scrub_event_in4</td>
<td>Input</td>
<td>1</td>
<td>Scrub event 4 trigger</td>
</tr>
<tr>
<td>scrub_event_in5</td>
<td>Input</td>
<td>1</td>
<td>Scrub event 5 trigger</td>
</tr>
<tr>
<td>scrub_event_in6</td>
<td>Input</td>
<td>1</td>
<td>Scrub event 6 trigger</td>
</tr>
</tbody>
</table>
### Table A-4  DMC Events list (continued)

<table>
<thead>
<tr>
<th>Signal</th>
<th>Type</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>scrub_event_in7</td>
<td>Input</td>
<td>1</td>
<td>Scrub event 7 trigger.</td>
</tr>
<tr>
<td>scrub_event_out0</td>
<td>Output</td>
<td>1</td>
<td>Scrub event 0 triggered.</td>
</tr>
<tr>
<td>scrub_event_out1</td>
<td>Output</td>
<td>1</td>
<td>Scrub event 1 triggered.</td>
</tr>
<tr>
<td>scrub_event_out2</td>
<td>Output</td>
<td>1</td>
<td>Scrub event 2 triggered.</td>
</tr>
<tr>
<td>scrub_event_out3</td>
<td>Output</td>
<td>1</td>
<td>Scrub event 3 triggered.</td>
</tr>
<tr>
<td>scrub_event_out4</td>
<td>Output</td>
<td>1</td>
<td>Scrub event 4 triggered.</td>
</tr>
<tr>
<td>scrub_event_out5</td>
<td>Output</td>
<td>1</td>
<td>Scrub event 5 triggered.</td>
</tr>
<tr>
<td>scrub_event_out6</td>
<td>Output</td>
<td>1</td>
<td>Scrub event 6 triggered.</td>
</tr>
<tr>
<td>scrub_event_out7</td>
<td>Output</td>
<td>1</td>
<td>Scrub event 7 triggered.</td>
</tr>
<tr>
<td>direct_cmd_event_in0</td>
<td>Input</td>
<td>1</td>
<td>Direct cmd event 0 trigger.</td>
</tr>
<tr>
<td>direct_cmd_event_in1</td>
<td>Input</td>
<td>1</td>
<td>Direct cmd event 1 trigger.</td>
</tr>
<tr>
<td>direct_cmd_event_in2</td>
<td>Input</td>
<td>1</td>
<td>Direct cmd event 2 trigger.</td>
</tr>
<tr>
<td>direct_cmd_event_in3</td>
<td>Input</td>
<td>1</td>
<td>Direct cmd event 3 trigger.</td>
</tr>
<tr>
<td>direct_cmd_event_out0</td>
<td>Output</td>
<td>1</td>
<td>Direct cmd event 0 triggered.</td>
</tr>
<tr>
<td>direct_cmd_event_out1</td>
<td>Output</td>
<td>1</td>
<td>Direct cmd event 1 triggered.</td>
</tr>
<tr>
<td>direct_cmd_event_out2</td>
<td>Output</td>
<td>1</td>
<td>Direct cmd event 2 triggered.</td>
</tr>
<tr>
<td>direct_cmd_event_out3</td>
<td>Output</td>
<td>1</td>
<td>Direct cmd event 3 triggered.</td>
</tr>
</tbody>
</table>

The following table shows the Scan Signals list of the DMC.

### Table A-5  DMC Scan Signals list

<table>
<thead>
<tr>
<th>Signal</th>
<th>Type</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dftse</td>
<td>Input</td>
<td>1</td>
<td>DFT scan enable</td>
</tr>
<tr>
<td>dftclkkgcen</td>
<td>Input</td>
<td>1</td>
<td>DFT clk clock gate enable</td>
</tr>
<tr>
<td>dftcldkdiv2cgen</td>
<td>Input</td>
<td>1</td>
<td>DFT clkdiv2 clock gate enable</td>
</tr>
<tr>
<td>dftrstvable</td>
<td>Input</td>
<td>1</td>
<td>DFT reset synchroniser disable</td>
</tr>
<tr>
<td>dftramhold</td>
<td>Input</td>
<td>1</td>
<td>DFT on-chip RAM hold</td>
</tr>
<tr>
<td>dfmephold</td>
<td>Input</td>
<td>1</td>
<td>DFT multi-cycle path hold</td>
</tr>
</tbody>
</table>

The following table shows the PMU Signals list of the DMC.

### Table A-6  DMC PMU Signals list

<table>
<thead>
<tr>
<th>Signal</th>
<th>Type</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ev_request_valid_valid</td>
<td>Output</td>
<td>1</td>
<td>Indicates that ev_request_valid_payload is valid</td>
</tr>
<tr>
<td>ev_request_tzfail_valid</td>
<td>Output</td>
<td>1</td>
<td>Indicates that ev_request_tzfail_payload is valid</td>
</tr>
<tr>
<td>ev_request_retry_valid</td>
<td>Output</td>
<td>1</td>
<td>Indicates that ev_request_retry_payload is valid</td>
</tr>
<tr>
<td>Signal</td>
<td>Type</td>
<td>Width</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ev_retry_grant_valid</td>
<td>Output</td>
<td>1</td>
<td>Indicates that ev_retry_grant_payload is valid</td>
</tr>
<tr>
<td>ev_request_valid_payload</td>
<td>Output</td>
<td>27</td>
<td>A request enters the DMC</td>
</tr>
<tr>
<td>ev_request_tzfail_payload</td>
<td>Output</td>
<td>22</td>
<td>A request fails an address translation or TrustZone permissions check</td>
</tr>
<tr>
<td>ev_request_retry_payload</td>
<td>Output</td>
<td>25</td>
<td>A request is retried</td>
</tr>
<tr>
<td>ev_retry_grant_payload</td>
<td>Output</td>
<td>15</td>
<td>Indicates that a P-credit has been granted.</td>
</tr>
<tr>
<td>ev_queue_fill_status_payload</td>
<td>Output</td>
<td>8</td>
<td>Count of entries in the DMC</td>
</tr>
<tr>
<td>ev_queued_reads_payload</td>
<td>Output</td>
<td>8</td>
<td>Count of read entries in the DMC</td>
</tr>
<tr>
<td>ev_queued_writes_payload</td>
<td>Output</td>
<td>8</td>
<td>Count of write entries in the DMC</td>
</tr>
<tr>
<td>ev_enqueued_reads_payload</td>
<td>Output</td>
<td>8</td>
<td>Count of read entries in the queue</td>
</tr>
<tr>
<td>ev_enqueued_writes_payload</td>
<td>Output</td>
<td>8</td>
<td>Count of write entries in the queue</td>
</tr>
<tr>
<td>ev_arbitrated_reads_payload</td>
<td>Output</td>
<td>8</td>
<td>Count of read entries in the arbitrated, without data state</td>
</tr>
<tr>
<td>ev_arbitrated_writes_payload</td>
<td>Output</td>
<td>8</td>
<td>Count of write entries in the arbitrated, not clean state</td>
</tr>
<tr>
<td>ev_read_backlog_payload</td>
<td>Output</td>
<td>8</td>
<td>Count of entries in the backlog queue</td>
</tr>
<tr>
<td>ev_enqueue_backlog_payload</td>
<td>Output</td>
<td>8</td>
<td>Count of entries that are waiting to get enqueued</td>
</tr>
<tr>
<td>ev_hazard_resolution_backlog_payload</td>
<td>Output</td>
<td>8</td>
<td>Count of entries that are ready to be merged</td>
</tr>
<tr>
<td>ev_queue_allocation_backlog_payload</td>
<td>Output</td>
<td>8</td>
<td>Count of entries that in allocation backlog</td>
</tr>
<tr>
<td>ev_enqueue_valid</td>
<td>Output</td>
<td>1</td>
<td>Indicates that ev_enqueue_payload is valid</td>
</tr>
<tr>
<td>ev_arbitrate_valid</td>
<td>Output</td>
<td>1</td>
<td>Indicates that ev_arbitrate_payload is valid</td>
</tr>
<tr>
<td>ev_rank_targetted_valid</td>
<td>Output</td>
<td>RANKS_PER_CHANNEL</td>
<td>A rank is targeted by an enqueued entry</td>
</tr>
<tr>
<td>ev_enqueue_payload</td>
<td>Output</td>
<td>34</td>
<td>A request is enqueued in the arbitration queue</td>
</tr>
<tr>
<td>ev_arbitrate_payload</td>
<td>Output</td>
<td>12</td>
<td>A request is arbitrated from the arbitration queue</td>
</tr>
<tr>
<td>ev_allocate_valid</td>
<td>Output</td>
<td>1</td>
<td>Indicates that ev_allocate_payload is valid</td>
</tr>
<tr>
<td>ev_allocate_payload</td>
<td>Output</td>
<td>22</td>
<td>Maps sysid to allocated tag ID on entry to the DCB</td>
</tr>
<tr>
<td>ev_request_hazard_valid</td>
<td>Output</td>
<td>1</td>
<td>Indicates that ev_request_hazard_payload is valid</td>
</tr>
<tr>
<td>ev_request_hazard_payload</td>
<td>Output</td>
<td>2</td>
<td>A request forms a data hazard on an existing entry</td>
</tr>
<tr>
<td>ev_request_partial_valid</td>
<td>Output</td>
<td>1</td>
<td>A request is partial (not a complete burst)</td>
</tr>
<tr>
<td>ev_request_rmw_valid</td>
<td>Output</td>
<td>1</td>
<td>A request requires a read-modify-write</td>
</tr>
<tr>
<td>ev_ram_err_detect_valid</td>
<td>Output</td>
<td>9</td>
<td>Indicates that ev_ram_err_detect_payload is valid</td>
</tr>
<tr>
<td>ev_ram_err_detect_payload</td>
<td>Output</td>
<td>42</td>
<td>See ram_ecc_errd_int description</td>
</tr>
<tr>
<td>ev_ram_err_correct_valid</td>
<td>Output</td>
<td>9</td>
<td>Indicates that ev_ram_err_correct_payload is valid</td>
</tr>
<tr>
<td>ev_ram_err_correct_payload</td>
<td>Output</td>
<td>42</td>
<td>See ram_ecc_errc_int description</td>
</tr>
<tr>
<td>ev_dram_err_detect_valid</td>
<td>Output</td>
<td>1</td>
<td>Indicates that ev_dram_err_detect_payload is valid</td>
</tr>
<tr>
<td>ev_dram_err_detect_payload</td>
<td>Output</td>
<td>35</td>
<td>See dram_ecc_errd_int description</td>
</tr>
</tbody>
</table>
### Table A-6  DMC PMU Signals list (continued)

<table>
<thead>
<tr>
<th>Signal</th>
<th>Type</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ev_dram_err_correct_valid</td>
<td>Output</td>
<td>1</td>
<td>Indicates that ev_dram_err_correct_payload is valid</td>
</tr>
<tr>
<td>ev_dram_err_correct_payload</td>
<td>Output</td>
<td>59</td>
<td>See dram_ecc_errc_int description</td>
</tr>
<tr>
<td>ev_turnaround_valid</td>
<td>Output</td>
<td>1</td>
<td>Indicates that ev_rank_turnaround_payload is valid</td>
</tr>
<tr>
<td>ev_activate_valid</td>
<td>Output</td>
<td>1</td>
<td>Indicates that ev_activate_payload is valid</td>
</tr>
<tr>
<td>ev_rdwr_valid</td>
<td>Output</td>
<td>1</td>
<td>Indicates that ev_rdwr_payload is valid</td>
</tr>
<tr>
<td>ev_precharge_valid</td>
<td>Output</td>
<td>1</td>
<td>Indicates that ev_precharge_payload is valid</td>
</tr>
<tr>
<td>ev_refresh_valid</td>
<td>Output</td>
<td>1</td>
<td>Indicates that ev_refresh_payload is valid</td>
</tr>
<tr>
<td>ev_turnaround_payload</td>
<td>Output</td>
<td>9</td>
<td>A turnaround has occurred</td>
</tr>
<tr>
<td>ev_activate_payload</td>
<td>Output</td>
<td>29</td>
<td>An ACTIVATE command has been sent</td>
</tr>
<tr>
<td>ev_rdwr_payload</td>
<td>Output</td>
<td>15</td>
<td>A READ/WRITE command has been sent</td>
</tr>
<tr>
<td>ev_precharge_payload</td>
<td>Output</td>
<td>9</td>
<td>A PRECHARGE command has been sent</td>
</tr>
<tr>
<td>ev_refresh_payload</td>
<td>Output</td>
<td>3</td>
<td>A REFRESH command has been sent</td>
</tr>
<tr>
<td>ev_pwr_state_active_valid</td>
<td>Output</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>The rank is active</td>
</tr>
<tr>
<td>ev_pwr_state_idle_valid</td>
<td>Output</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>The rank is idle</td>
</tr>
<tr>
<td>ev_pwr_state_pd_valid</td>
<td>Output</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>The rank is in a POWER DOWN state</td>
</tr>
<tr>
<td>ev_pwr_state_sref_valid</td>
<td>Output</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>The rank is in a SELF_REFRESH state</td>
</tr>
<tr>
<td>ev_bank_active_valid</td>
<td>Output</td>
<td>BANKS_PER_CHANNEL</td>
<td>A bank is active (has a row open)</td>
</tr>
<tr>
<td>ev_bank_busy_valid</td>
<td>Output</td>
<td>BANKS_PER_CHANNEL</td>
<td>A bank is busy (one or more timing parameters is being measured following an access)</td>
</tr>
<tr>
<td>ev_phy_update_req_valid</td>
<td>Output</td>
<td>1</td>
<td>Indicates that ev_phy_update_req_payload is valid</td>
</tr>
<tr>
<td>ev_phy_update_valid</td>
<td>Output</td>
<td>1</td>
<td>Indicates that ev_phy_update_payload is valid</td>
</tr>
<tr>
<td>ev_phy_update_req_payload</td>
<td>Output</td>
<td>4</td>
<td>A PHY update request has been received (update or training)</td>
</tr>
<tr>
<td>ev_phy_update_payload</td>
<td>Output</td>
<td>4</td>
<td>A PHY update request is in progress (update or training)</td>
</tr>
<tr>
<td>ev_phy_update_complete_valid</td>
<td>Output</td>
<td>1</td>
<td>A PHY update request has been completed (update or training)</td>
</tr>
<tr>
<td>ev_link_err_valid</td>
<td>Output</td>
<td>1</td>
<td>A link error has been detected</td>
</tr>
<tr>
<td>ev_tmac_limit_reached_valid</td>
<td>Output</td>
<td>1</td>
<td>Indicates that a bank row has reached the tMAC threshold for triggering a Target Row Refresh</td>
</tr>
<tr>
<td>ev_tmacaw_tracker_full_valid</td>
<td>Output</td>
<td>1</td>
<td>Indicates that tMAC/tMAW tracking resource is full</td>
</tr>
</tbody>
</table>

The following table shows the Misc. signals list of the DMC.

### Table A-7  DMC Misc. signals list

<table>
<thead>
<tr>
<th>Signal</th>
<th>Type</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>memory_type</td>
<td>Output</td>
<td>3</td>
<td>An external output of the value of the memory_type register bitfield.</td>
</tr>
</tbody>
</table>
The following table shows the Tie-off signals list of the DMC.

**Table A-8**  DMC Tie-off signals list

<table>
<thead>
<tr>
<th>Signal</th>
<th>Type</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>t_rddata_en_diff_tie_off</td>
<td>Input</td>
<td>6</td>
<td>Tie-off value for reset of register bitfield t_rddata_en_diff</td>
</tr>
<tr>
<td>t_phyrdcslat_tie_off</td>
<td>Input</td>
<td>5</td>
<td>Tie-off value for reset of register bitfield t_phyrdcslat</td>
</tr>
<tr>
<td>t_phyrdlat_tie_off</td>
<td>Input</td>
<td>7</td>
<td>Tie-off value for reset of register bitfield t_phyrdlat</td>
</tr>
<tr>
<td>t_phywrлат_diff_tie_off</td>
<td>Input</td>
<td>5</td>
<td>Tie-off value for reset of register bitfield t_phywrлат_diff</td>
</tr>
<tr>
<td>t_phywrcslat_tie_off</td>
<td>Input</td>
<td>5</td>
<td>Tie-off value for reset of register bitfield t_phywrcslat</td>
</tr>
<tr>
<td>t_phywrdata_tie_off</td>
<td>Input</td>
<td>1</td>
<td>Tie-off value for reset of register bitfield t_phywrdata</td>
</tr>
<tr>
<td>refresh_dur_rdlvl_tie_off</td>
<td>Input</td>
<td>1</td>
<td>Tie-off value for reset of register bitfield refresh_dur_rdlvl</td>
</tr>
<tr>
<td>t_rdlvl_en_tie_off</td>
<td>Input</td>
<td>6</td>
<td>Tie-off value for reset of register bitfield t_rdlvl_en</td>
</tr>
<tr>
<td>t_rdlvl_rr_tie_off</td>
<td>Input</td>
<td>10</td>
<td>Tie-off value for reset of register bitfield t_rdlvl_rr</td>
</tr>
<tr>
<td>refresh_dur_wrvl_tie_off</td>
<td>Input</td>
<td>1</td>
<td>Tie-off value for reset of register bitfield refresh_dur_wrvl</td>
</tr>
<tr>
<td>t_wrvl_en_tie_off</td>
<td>Input</td>
<td>6</td>
<td>Tie-off value for reset of register bitfield t_wrvl_en</td>
</tr>
<tr>
<td>t_wrvl_ww_tie_off</td>
<td>Input</td>
<td>10</td>
<td>Tie-off value for reset of register bitfield t_wrvl_ww</td>
</tr>
<tr>
<td>t_lpresp_tie_off</td>
<td>Input</td>
<td>6</td>
<td>Tie-off value for reset of register bitfield t_lpresp</td>
</tr>
<tr>
<td>user_config0_tie_off</td>
<td>Input</td>
<td>32</td>
<td>Tie-off value for reset of register bitfield user_config0</td>
</tr>
<tr>
<td>user_config1_tie_off</td>
<td>Input</td>
<td>32</td>
<td>Tie-off value for reset of register bitfield user_config1</td>
</tr>
<tr>
<td>user_config2_tie_off</td>
<td>Input</td>
<td>32</td>
<td>Tie-off value for reset of register bitfield user_config2</td>
</tr>
<tr>
<td>user_config3_tie_off</td>
<td>Input</td>
<td>32</td>
<td>Tie-off value for reset of register bitfield user_config3</td>
</tr>
</tbody>
</table>

The following table shows the Tie-off values for AMBA5 CHI list of the DMC.

**Table A-9**  DMC Tie-off values for AMBA5 CHI list

<table>
<thead>
<tr>
<th>Signal</th>
<th>Type</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>system_id</td>
<td>Input</td>
<td>SKY_RSP_FLIT_SRCID_WIDTH</td>
<td>Tie-off value to set the physical node ID of the DMC</td>
</tr>
<tr>
<td>home_node_id</td>
<td>Input</td>
<td>(SKY_REQ_FLIT_SRCID_WIDTH*SYSTEM_REQUESTORS)</td>
<td>Tie-off value to specify the concatenated physical node IDs of up to 8 Home Nodes that are permitted to access the DMC</td>
</tr>
</tbody>
</table>

The following table shows the APB Interface bus list of the DMC.

**Table A-10**  DMC APB Interface list

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>paddr</td>
<td>32</td>
<td>APB address</td>
</tr>
<tr>
<td>psel</td>
<td></td>
<td>APB select</td>
</tr>
<tr>
<td>penable</td>
<td></td>
<td>APB enable</td>
</tr>
<tr>
<td>pwrite</td>
<td></td>
<td>APB write</td>
</tr>
<tr>
<td>pwdata</td>
<td>32</td>
<td>APB write data</td>
</tr>
</tbody>
</table>
The following table shows the DFI Interface bus list of the DMC.

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dfi_address_p0</td>
<td>18</td>
<td>Address to DDR3 PHY</td>
</tr>
<tr>
<td>dfi_address_p1</td>
<td>18</td>
<td>Address to DDR3 PHY</td>
</tr>
<tr>
<td>dfi_address_p2</td>
<td>18</td>
<td>Address to DDR3 PHY</td>
</tr>
<tr>
<td>dfi_address_p3</td>
<td>18</td>
<td>Address to DDR3 PHY</td>
</tr>
<tr>
<td>dfi_bank_p0</td>
<td>3</td>
<td>Bank Address to PHY</td>
</tr>
<tr>
<td>dfi_bank_p1</td>
<td>3</td>
<td>Bank Address to PHY</td>
</tr>
<tr>
<td>dfi_bank_p2</td>
<td>3</td>
<td>Bank Address to PHY</td>
</tr>
<tr>
<td>dfi_bank_p3</td>
<td>3</td>
<td>Bank Address to PHY</td>
</tr>
<tr>
<td>dfi_ras_n_p0</td>
<td>1</td>
<td>Row address strobe to PHY</td>
</tr>
<tr>
<td>dfi_ras_n_p1</td>
<td>1</td>
<td>Row address strobe to PHY</td>
</tr>
<tr>
<td>dfi_ras_n_p2</td>
<td>1</td>
<td>Row address strobe to PHY</td>
</tr>
<tr>
<td>dfi_ras_n_p3</td>
<td>1</td>
<td>Row address strobe to PHY</td>
</tr>
<tr>
<td>dfiCas_n_p0</td>
<td>1</td>
<td>Column address strobe to PHY</td>
</tr>
<tr>
<td>dfiCas_n_p1</td>
<td>1</td>
<td>Column address strobe to PHY</td>
</tr>
<tr>
<td>dfiCas_n_p2</td>
<td>1</td>
<td>Column address strobe to PHY</td>
</tr>
<tr>
<td>dfiCas_n_p3</td>
<td>1</td>
<td>Column address strobe to PHY</td>
</tr>
<tr>
<td>dfi_we_n_p0</td>
<td>1</td>
<td>Write enable to PHY</td>
</tr>
<tr>
<td>dfi_we_n_p1</td>
<td>1</td>
<td>Write enable to PHY</td>
</tr>
<tr>
<td>dfi_we_n_p2</td>
<td>1</td>
<td>Write enable to PHY</td>
</tr>
<tr>
<td>dfi_we_n_p3</td>
<td>1</td>
<td>Write enable to PHY</td>
</tr>
<tr>
<td>dfi_cs_n_p0</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>Chip-select to PHY</td>
</tr>
<tr>
<td>dfi_cs_n_p1</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>Chip-select to PHY</td>
</tr>
<tr>
<td>dfi_cs_n_p2</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>Chip-select to PHY</td>
</tr>
<tr>
<td>dfi_cs_n_p3</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>Chip-select to PHY</td>
</tr>
<tr>
<td>dfi_act_n_p0</td>
<td>1</td>
<td>Activate to PHY</td>
</tr>
<tr>
<td>dfi_act_n_p1</td>
<td>1</td>
<td>Activate to PHY</td>
</tr>
<tr>
<td>dfi_act_n_p2</td>
<td>1</td>
<td>Activate to PHY</td>
</tr>
<tr>
<td>dfi_act_n_p3</td>
<td>1</td>
<td>Activate to PHY</td>
</tr>
<tr>
<td>Name</td>
<td>Width</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>dfi_bg_p0</td>
<td>2</td>
<td>Bank group address to PHY</td>
</tr>
<tr>
<td>dfi_bg_p1</td>
<td>2</td>
<td>Bank group address to PHY</td>
</tr>
<tr>
<td>dfi_bg_p2</td>
<td>2</td>
<td>Bank group address to PHY</td>
</tr>
<tr>
<td>dfi_bg_p3</td>
<td>2</td>
<td>Bank group address to PHY</td>
</tr>
<tr>
<td>dfi_cid_p0</td>
<td>3</td>
<td>Chip ID to PHY</td>
</tr>
<tr>
<td>dfi_cid_p1</td>
<td>3</td>
<td>Chip ID to PHY</td>
</tr>
<tr>
<td>dfi_cid_p2</td>
<td>3</td>
<td>Chip ID to PHY</td>
</tr>
<tr>
<td>dfi_cid_p3</td>
<td>3</td>
<td>Chip ID to PHY</td>
</tr>
<tr>
<td>dfi_cke_p0</td>
<td></td>
<td>Clock enable to PHY</td>
</tr>
<tr>
<td>dfi_cke_p1</td>
<td></td>
<td>Clock enable to PHY</td>
</tr>
<tr>
<td>dfi_cke_p2</td>
<td></td>
<td>Clock enable to PHY</td>
</tr>
<tr>
<td>dfi_cke_p3</td>
<td></td>
<td>Clock enable to PHY</td>
</tr>
<tr>
<td>dfi_odt_p0</td>
<td></td>
<td>On Die Termination to PHY</td>
</tr>
<tr>
<td>dfi_odt_p1</td>
<td></td>
<td>On Die Termination to PHY</td>
</tr>
<tr>
<td>dfi_odt_p2</td>
<td></td>
<td>On Die Termination to PHY</td>
</tr>
<tr>
<td>dfi_odt_p3</td>
<td></td>
<td>On Die Termination to PHY</td>
</tr>
<tr>
<td>dfi_reset_n_p0</td>
<td></td>
<td>Reset to PHY</td>
</tr>
<tr>
<td>dfi_reset_n_p1</td>
<td></td>
<td>Reset to PHY</td>
</tr>
<tr>
<td>dfi_reset_n_p2</td>
<td></td>
<td>Reset to PHY</td>
</tr>
<tr>
<td>dfi_reset_n_p3</td>
<td></td>
<td>Reset to PHY</td>
</tr>
<tr>
<td>dfi_parity_in_p0</td>
<td>1</td>
<td>Command parity to PHY</td>
</tr>
<tr>
<td>dfi_parity_in_p1</td>
<td>1</td>
<td>Command parity to PHY</td>
</tr>
<tr>
<td>dfi_parity_in_p2</td>
<td>1</td>
<td>Command parity to PHY</td>
</tr>
<tr>
<td>dfi_parity_in_p3</td>
<td>1</td>
<td>Command parity to PHY</td>
</tr>
<tr>
<td>dfi_wrdata_en_p0</td>
<td>(DFI_DATA_SLICES)</td>
<td>Write data enable PHY</td>
</tr>
<tr>
<td>dfi_wrdata_en_p1</td>
<td>(DFI_DATA_SLICES)</td>
<td>Write data enable PHY</td>
</tr>
<tr>
<td>dfi_wrdata_en_p2</td>
<td>(DFI_DATA_SLICES)</td>
<td>Write data enable PHY</td>
</tr>
<tr>
<td>dfi_wrdata_en_p3</td>
<td>(DFI_DATA_SLICES)</td>
<td>Write data enable PHY</td>
</tr>
<tr>
<td>dfi_wrdata_p0</td>
<td>(DFI_DATA_BITS)</td>
<td>Write data to PHY</td>
</tr>
<tr>
<td>dfi_wrdata_p1</td>
<td>(DFI_DATA_BITS)</td>
<td>Write data to PHY</td>
</tr>
<tr>
<td>dfi_wrdata_p2</td>
<td>(DFI_DATA_BITS)</td>
<td>Write data to PHY</td>
</tr>
<tr>
<td>dfi_wrdata_p3</td>
<td>(DFI_DATA_BITS)</td>
<td>Write data to PHY</td>
</tr>
<tr>
<td>dfi_wrdata_cs_n_p0</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>Write Data Path Chip-select to PHY</td>
</tr>
<tr>
<td>dfi_wrdata_cs_n_p1</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>Write Data Path Chip-select to PHY</td>
</tr>
<tr>
<td>dfi_wrdata_cs_n_p2</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>Write Data Path Chip-select to PHY</td>
</tr>
<tr>
<td>Name</td>
<td>Width</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------</td>
<td>------------------------------------------------------------</td>
</tr>
<tr>
<td>dfi_wrdata_cs_n_p3</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>Write Data Path Chip-select to PHY</td>
</tr>
<tr>
<td>dfi_wrdata_mask_p0</td>
<td>(DFI_DATA_BYTES)</td>
<td>Write data mask PHY</td>
</tr>
<tr>
<td>dfi_wrdata_mask_p1</td>
<td>(DFI_DATA_BYTES)</td>
<td>Write data mask PHY</td>
</tr>
<tr>
<td>dfi_wrdata_mask_p2</td>
<td>(DFI_DATA_BYTES)</td>
<td>Write data mask PHY</td>
</tr>
<tr>
<td>dfi_wrdata_mask_p3</td>
<td>(DFI_DATA_BYTES)</td>
<td>Write data mask PHY</td>
</tr>
<tr>
<td>dfi_rddata_en_p0</td>
<td>(DFI_DATA_SLICES)</td>
<td>Enable for read data</td>
</tr>
<tr>
<td>dfi_rddata_en_p1</td>
<td>(DFI_DATA_SLICES)</td>
<td>Enable for read data</td>
</tr>
<tr>
<td>dfi_rddata_en_p2</td>
<td>(DFI_DATA_SLICES)</td>
<td>Enable for read data</td>
</tr>
<tr>
<td>dfi_rddata_en_p3</td>
<td>(DFI_DATA_SLICES)</td>
<td>Enable for read data</td>
</tr>
<tr>
<td>dfi_rddata_p0</td>
<td>(DFI_DATA_BITS)</td>
<td>Read data input from PHY</td>
</tr>
<tr>
<td>dfi_rddata_p1</td>
<td>(DFI_DATA_BITS)</td>
<td>Read data input from PHY</td>
</tr>
<tr>
<td>dfi_rddata_p2</td>
<td>(DFI_DATA_BITS)</td>
<td>Read data input from PHY</td>
</tr>
<tr>
<td>dfi_rddata_p3</td>
<td>(DFI_DATA_BITS)</td>
<td>Read data input from PHY</td>
</tr>
<tr>
<td>dfi_rddata_dbi_n_p0</td>
<td>(DFI_DATA_BYTES) * 2</td>
<td>Read Data DBI. This signal is sent with dfi_rddata bus indicating DBI functionality. If not used this signal should be tied to 'b1.</td>
</tr>
<tr>
<td>dfi_rddata_dbi_n_p1</td>
<td>(DFI_DATA_BYTES) * 2</td>
<td>Read Data DBI. This signal is sent with dfi_rddata bus indicating DBI functionality. If not used this signal should be tied to 'b1.</td>
</tr>
<tr>
<td>dfi_rddata_dbi_n_p2</td>
<td>(DFI_DATA_BYTES) * 2</td>
<td>Read Data DBI. This signal is sent with dfi_rddata bus indicating DBI functionality. If not used this signal should be tied to 'b1.</td>
</tr>
<tr>
<td>dfi_rddata_dbi_n_p3</td>
<td>(DFI_DATA_BYTES) * 2</td>
<td>Read Data DBI. This signal is sent with dfi_rddata bus indicating DBI functionality. If not used this signal should be tied to 'b1.</td>
</tr>
<tr>
<td>dfi_rddata_valid_p0</td>
<td>(DFI_DATA_SLICES)</td>
<td>Indicates read data valid</td>
</tr>
<tr>
<td>dfi_rddata_valid_p1</td>
<td>(DFI_DATA_SLICES)</td>
<td>Indicates read data valid</td>
</tr>
<tr>
<td>dfi_rddata_valid_p2</td>
<td>(DFI_DATA_SLICES)</td>
<td>Indicates read data valid</td>
</tr>
<tr>
<td>dfi_rddata_valid_p3</td>
<td>(DFI_DATA_SLICES)</td>
<td>Indicates read data valid</td>
</tr>
<tr>
<td>dfi_rddata_cs_n_p0</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>Read Data Path Chip-select to PHY</td>
</tr>
<tr>
<td>dfi_rddata_cs_n_p1</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>Read Data Path Chip-select to PHY</td>
</tr>
<tr>
<td>dfi_rddata_cs_n_p2</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>Read Data Path Chip-select to PHY</td>
</tr>
<tr>
<td>dfi_rddata_cs_n_p3</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>Read Data Path Chip-select to PHY</td>
</tr>
<tr>
<td>dfi_ctrlupd_req</td>
<td>1</td>
<td>This signal is part of DFI 3.0, see JEDEC specification for details.</td>
</tr>
<tr>
<td>dfi_ctrlupd_ack</td>
<td>1</td>
<td>This signal is part of DFI 3.0, see JEDEC specification for details.</td>
</tr>
<tr>
<td>dfi_phyupd_req</td>
<td>1</td>
<td>DFI PHY-initiated update request</td>
</tr>
<tr>
<td>dfi_phyupd_ack</td>
<td>1</td>
<td>DFI PHY-initiated update acknowledge</td>
</tr>
<tr>
<td>dfi_phyupd_type</td>
<td>2</td>
<td>DFI PHY-initiated update type</td>
</tr>
<tr>
<td>dfi_data_byte_disable</td>
<td>(DFI_DATA_BYTES)</td>
<td>This signal is part of DFI 3.0, see JEDEC specification for details.</td>
</tr>
<tr>
<td>dfi_dram_clk_disable</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>DRAM clock disable to PHY</td>
</tr>
<tr>
<td>Name</td>
<td>Width</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>dfi_init_start</td>
<td>1</td>
<td>This signal is part of DFI 3.0, see JEDEC specification for details.</td>
</tr>
<tr>
<td>dfi_init_complete</td>
<td>1</td>
<td>Indicates PHY initialisation complete</td>
</tr>
<tr>
<td>dfi_alert_n_p0</td>
<td>1</td>
<td>This signal is part of DFI 3.0, see JEDEC specification for details.</td>
</tr>
<tr>
<td>dfi_alert_n_p1</td>
<td>1</td>
<td>This signal is part of DFI 3.0, see JEDEC specification for details.</td>
</tr>
<tr>
<td>dfi_alert_n_p2</td>
<td>1</td>
<td>This signal is part of DFI 3.0, see JEDEC specification for details.</td>
</tr>
<tr>
<td>dfi_alert_n_p3</td>
<td>1</td>
<td>This signal is part of DFI 3.0, see JEDEC specification for details.</td>
</tr>
<tr>
<td>dfi_err</td>
<td>1</td>
<td>This signal is part of DFI 3.0, see JEDEC specification for details.</td>
</tr>
<tr>
<td>dfi_err_info</td>
<td>4</td>
<td>This signal is part of DFI 3.0, see JEDEC specification for details.</td>
</tr>
<tr>
<td>dfi_phylvl_req_cs_n</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>This signal is part of DFI 3.0, see JEDEC specification for details.</td>
</tr>
<tr>
<td>dfi_phylvl_ack_cs_n</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>This signal is part of DFI 3.0, see JEDEC specification for details.</td>
</tr>
<tr>
<td>dfi_rdlvl_req</td>
<td>1</td>
<td>DFI read data eye training request</td>
</tr>
<tr>
<td>dfi_rdlvl_cs_n</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>DFI read data eye training request target chip-select</td>
</tr>
<tr>
<td>dfi_rdlvl_periodic</td>
<td>1</td>
<td>DFI read data eye training request periodic</td>
</tr>
<tr>
<td>dfi_rdlvl_en</td>
<td>1</td>
<td>DFI read data eye training enable</td>
</tr>
<tr>
<td>dfi_rdlvl_resp</td>
<td>1</td>
<td>DFI read data eye training response</td>
</tr>
<tr>
<td>dfi_rdlvl_gate_req</td>
<td>1</td>
<td>DFI read gate training request</td>
</tr>
<tr>
<td>dfi_rdlvl_gate_cs_n</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>DFI read gate training request target chip-select</td>
</tr>
<tr>
<td>dfi_rdlvl_gate_periodic</td>
<td>1</td>
<td>DFI read gate training request periodic</td>
</tr>
<tr>
<td>dfi_rdlvl_gate_en</td>
<td>1</td>
<td>DFI read gate training enable</td>
</tr>
<tr>
<td>dfi_rdlvl_gate_resp</td>
<td>1</td>
<td>DFI read gate training response</td>
</tr>
<tr>
<td>dfi_wrlvl_req</td>
<td>1</td>
<td>DFI write leveling training request</td>
</tr>
<tr>
<td>dfi_wrlvl_cs_n</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>DFI write leveling training request target chip-select</td>
</tr>
<tr>
<td>dfi_wrlvl_periodic</td>
<td>1</td>
<td>DFI write leveling training request periodic</td>
</tr>
<tr>
<td>dfi_wrlvl_en</td>
<td>1</td>
<td>DFI write leveling training enable</td>
</tr>
<tr>
<td>dfi_wrlvl_strobe</td>
<td>1</td>
<td>DFI write leveling training strobe</td>
</tr>
<tr>
<td>dfi_wrlvl_resp</td>
<td>1</td>
<td>DFI write leveling training response</td>
</tr>
<tr>
<td>dfi_lvl_pattern</td>
<td>4</td>
<td>This signal is part of DFI 3.0, see JEDEC specification for details.</td>
</tr>
<tr>
<td>dfi_lvl_periodic</td>
<td>1</td>
<td>This signal is part of DFI 3.0, see JEDEC specification for details.</td>
</tr>
<tr>
<td>dfi_lvl_cs_n</td>
<td>MEMORY_CHIP_SELECTS</td>
<td>This signal is part of DFI 3.0, see JEDEC specification for details.</td>
</tr>
<tr>
<td>dfi_ref_en</td>
<td>1</td>
<td>DFI refresh during training enable</td>
</tr>
<tr>
<td>dfi_lp_ctrl_req</td>
<td>1</td>
<td>DFI command low power request</td>
</tr>
<tr>
<td>dfi_lp_data_req</td>
<td>1</td>
<td>DFI data low power request</td>
</tr>
<tr>
<td>dfi_lp_wakeup</td>
<td>4</td>
<td>DFI command low power PHY wakeup allowance</td>
</tr>
<tr>
<td>dfi_lp_ack</td>
<td>1</td>
<td>DFI command low power acknowledge</td>
</tr>
</tbody>
</table>
The following table shows the Q-Channel Interface for DMC bus list of the DMC.

### Table A-12 DMC Q-Channel Interface for DMC list

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>qreqn</td>
<td>1</td>
<td>Request from the external clock controller to prepare to stop the clock</td>
</tr>
<tr>
<td>qacceptn</td>
<td>1</td>
<td>Positive acknowledgement after receiving QREQn assertion indicating that the DMC has completed preparation to stop the clocks and that the external clock controller can stop the clock</td>
</tr>
<tr>
<td>qdeny</td>
<td>1</td>
<td>Negative acknowledgement after receiving QREQn assertion indicating that the DMC has refused the request from the external clock controller to prepare to stop the clock</td>
</tr>
<tr>
<td>qactive</td>
<td>1</td>
<td>Indication that the DMC is active</td>
</tr>
</tbody>
</table>

The following table shows the Q-Channel Interface for APB interface bus list of the DMC.

### Table A-13 DMC Q-Channel Interface for APB interface list

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>qreqn_apb</td>
<td>1</td>
<td>Request from the external clock controller to prepare to stop the clock</td>
</tr>
<tr>
<td>qacceptn_apb</td>
<td>1</td>
<td>Positive acknowledgement after receiving QREQn assertion indicating that the APB interface has completed preparation to stop the clocks and that the external clock controller can stop the clock</td>
</tr>
<tr>
<td>qdeny_apb</td>
<td>1</td>
<td>Negative acknowledgement after receiving QREQn assertion indicating that the APB interface has refused the request from the external clock controller to prepare to stop the clock</td>
</tr>
<tr>
<td>qactive_apb</td>
<td>1</td>
<td>Indication that the APB interface is active</td>
</tr>
</tbody>
</table>

The following table shows the Clock Frequency Change Interface bus list of the DMC.

### Table A-14 DMC Clock Frequency Change Interface list

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cc_frequency</td>
<td>5</td>
<td>Used to indicate new frequency as part of frequency change protocol</td>
</tr>
<tr>
<td>cc_freq_change_req</td>
<td>1</td>
<td>Signals to an external clock control that the clock frequency can be updated</td>
</tr>
<tr>
<td>cc_freq_change_ack</td>
<td>1</td>
<td>Signals to the DMC from an external clock control that the clock frequency has been updated</td>
</tr>
</tbody>
</table>

The following table shows the Clock Frequency Change Interface bus list of the DMC.

### Table A-15 DMC Clock Frequency Change Interface list

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dfi_frequency</td>
<td>5</td>
<td>Used to indicate new frequency as part of frequency change protocol</td>
</tr>
<tr>
<td>dfi_freq_change_req</td>
<td>1</td>
<td>Signals to an external clock control that the clock frequency can be updated</td>
</tr>
<tr>
<td>dfi_freq_change_ack</td>
<td>1</td>
<td>Signals to the DMC from an external clock control that the clock frequency has been updated</td>
</tr>
</tbody>
</table>

The following table shows the Abort Interface bus list of the DMC.
The following table shows the Memory BIST interface bus list of the DMC.

**Table A-17 DMC Memory BIST interface list**

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mbistresetn</td>
<td>1</td>
<td>MBIST reset. Active low.</td>
</tr>
<tr>
<td>mbistreq</td>
<td>1</td>
<td>MBIST request</td>
</tr>
<tr>
<td>mbistack</td>
<td>1</td>
<td>MBIST acknowledge</td>
</tr>
<tr>
<td>mbistwriteen</td>
<td>1</td>
<td>MBIST write enable</td>
</tr>
<tr>
<td>mbistreaden</td>
<td>1</td>
<td>MBIST read enable</td>
</tr>
<tr>
<td>mbistaddr</td>
<td>7</td>
<td>MBIST address</td>
</tr>
<tr>
<td>mbistarray</td>
<td>5</td>
<td>MBIST array selection</td>
</tr>
<tr>
<td>mbistindata</td>
<td>154</td>
<td>MBIST write data</td>
</tr>
<tr>
<td>mbistoutdata</td>
<td>154</td>
<td>MBIST read data</td>
</tr>
</tbody>
</table>

The following table shows the Interrupt Interface bus list of the DMC.

**Table A-18 DMC Interrupt Interface list**

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ram_ecc_errc_int</td>
<td>1</td>
<td>The DMC has detected a correctable error in an internal RAM</td>
</tr>
</tbody>
</table>

The following table shows the Interrupt Interface bus list of the DMC.

**Table A-19 DMC Interrupt Interface list**

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ram_ecc_errd_int</td>
<td>1</td>
<td>The DMC has detected an un-correctable error in an internal RAM</td>
</tr>
</tbody>
</table>

The following table shows the Interrupt Interface bus list of the DMC.

**Table A-20 DMC Interrupt Interface list**

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dram_ecc_errc_int</td>
<td>1</td>
<td>The DMC has detected a correctable error in a DRAM burst</td>
</tr>
</tbody>
</table>

The following table shows the Interrupt Interface bus list of the DMC.

**Table A-21 DMC Interrupt Interface list**

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dram_ecc_errd_int</td>
<td>1</td>
<td>The DMC has detected a data failure that could not be corrected in a DRAM burst operation</td>
</tr>
</tbody>
</table>
The following table shows the Interrupt Interface bus list of the DMC.

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>failed_access_int</td>
<td>1</td>
<td>The DMC has detected a system request that has failed a permissions check</td>
</tr>
</tbody>
</table>

The following table shows the Interrupt Interface bus list of the DMC.

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>failed_prog_int</td>
<td>1</td>
<td>The DMC has detected a programming request that is not permitted</td>
</tr>
</tbody>
</table>

The following table shows the Interrupt Interface bus list of the DMC.

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>link_err_int</td>
<td>1</td>
<td>The DRAM interface has suffered from a link failure and a recovery attempt has begun</td>
</tr>
</tbody>
</table>

The following table shows the Interrupt Interface bus list of the DMC.

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>temperature_event_int</td>
<td>1</td>
<td>The DMC has detected a temperature event signalled by the DRAM, either directly, or if a temperature delta has been observed through automated polling of the temperature sensor</td>
</tr>
</tbody>
</table>

The following table shows the Interrupt Interface bus list of the DMC.

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>arch_fsm_int</td>
<td>1</td>
<td>The DMC has detected a change in the architectural state.</td>
</tr>
</tbody>
</table>

The following table shows the Interrupt Interface bus list of the DMC.

| Name          | Width | Description                 |
|---------------|-------|----------------------------|---|
| phy_request_int | 1     | The DMC has detected a PHY request. |

The following table shows the Interrupt Interface bus list of the DMC.

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>combined_int</td>
<td>1</td>
<td>A combined interrupt that is the logical OR of the other interrupts.</td>
</tr>
</tbody>
</table>
The following table shows the Interrupt Overflow bus list of the DMC.

### Table A-29 DMC Interrupt Overflow list

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ram_ecc_errc_oflow</td>
<td>1</td>
<td>The DMC has detected a correctable error in an internal RAM and a previously detected assertion was not cleared.</td>
</tr>
</tbody>
</table>

### Table A-30 DMC Interrupt Overflow list

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ram_ecc_errd_oflow</td>
<td>1</td>
<td>The DMC has detected an un-correctable error in an internal RAM and a previously detected assertion was not cleared.</td>
</tr>
</tbody>
</table>

### Table A-31 DMC Interrupt Overflow list

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dram_ecc_errc_oflow</td>
<td>1</td>
<td>The DMC has detected a correctable error in a DRAM burst and a previously detected assertion was not cleared.</td>
</tr>
</tbody>
</table>

### Table A-32 DMC Interrupt Overflow list

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dram_ecc_errd_oflow</td>
<td>1</td>
<td>The DMC has detected a data failure that could not be corrected in a DRAM burst operation and a previously detected assertion was not cleared.</td>
</tr>
</tbody>
</table>

### Table A-33 DMC Interrupt Overflow list

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>failed_access_oflow</td>
<td>1</td>
<td>The DMC has detected a system request that has failed a permissions check and a previously detected assertion was not cleared.</td>
</tr>
</tbody>
</table>

### Table A-34 DMC Interrupt Overflow list

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>failed_prog_oflow</td>
<td>1</td>
<td>The DMC has detected a programming request that is not permitted and a previously detected assertion was not cleared.</td>
</tr>
</tbody>
</table>
Table A-35  DMC Interrupt Overflow list

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>link_err_oflow</td>
<td>1</td>
<td>The DRAM interface has suffered from a link failure and a recovery attempt has begun and a previously detected assertion was not cleared.</td>
</tr>
</tbody>
</table>

The following table shows the Interrupt Overflow bus list of the DMC.

Table A-36  DMC Interrupt Overflow list

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>temperature_event_oflow</td>
<td>1</td>
<td>The DMC has detected a temperature event signalled by the DRAM, either directly, or if a temperature delta has been observed through automated polling of the temperature sensor and a previously detected assertion was not cleared.</td>
</tr>
</tbody>
</table>

The following table shows the Interrupt Overflow bus list of the DMC.

Table A-37  DMC Interrupt Overflow list

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>arch_fsm_oflow</td>
<td>1</td>
<td>The DMC has detected a change in the architectural state and a previously detected assertion was not cleared.</td>
</tr>
</tbody>
</table>

The following table shows the Interrupt Overflow bus list of the DMC.

Table A-38  DMC Interrupt Overflow list

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>phy_request_oflow</td>
<td>1</td>
<td>The DMC has detected a PHY request and a previously detected assertion was not cleared.</td>
</tr>
</tbody>
</table>

The following table shows the Interrupt Overflow bus list of the DMC.

Table A-39  DMC Interrupt Overflow list

<table>
<thead>
<tr>
<th>Name</th>
<th>Width</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>combined_oflow</td>
<td>1</td>
<td>A combined interrupt that is the logical OR of the other interrupt overflows.</td>
</tr>
</tbody>
</table>
Appendix B
Revisions

This appendix describes the technical changes between released issues of this book.

It contains the following sections:
•  B.1 Revisions on page Appx-B-161.
B.1 Revisions

This appendix describes the technical changes between released issues of this book.

### Table B-1 Revision 0000 issue 00

<table>
<thead>
<tr>
<th>Change</th>
<th>Location</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>First release</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

### Table B-2 Differences between revision 0000 issue 00 and revision 0001 issue 00

<table>
<thead>
<tr>
<th>Change</th>
<th>Location</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programmable registers updated</td>
<td>3.2 Register summary on page 3-33</td>
<td>All revisions</td>
</tr>
<tr>
<td>DIMM supported list updated</td>
<td>2.4 Constraints and limitations of use on page 2-29</td>
<td>All revisions</td>
</tr>
</tbody>
</table>

### Table B-3 Differences between revision 0001 issue 00 and revision 0100 issue 00

<table>
<thead>
<tr>
<th>Change</th>
<th>Location</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Added 32-bit memory width support to the memory interface.</td>
<td>1.3 Features on page 1-14</td>
<td>Revision r1p0</td>
</tr>
<tr>
<td>Added the format control register.</td>
<td>3.2 Register summary on page 3-33</td>
<td>Revision r1p0</td>
</tr>
<tr>
<td></td>
<td>3.3.6 format_control on page 3-46</td>
<td></td>
</tr>
</tbody>
</table>

### Table B-4 Differences between revision 0100 issue 00 and revision 0200 issue 00

<table>
<thead>
<tr>
<th>Change</th>
<th>Location</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Updated the Memory interface support.</td>
<td>1.3 Features on page 1-14</td>
<td>Revision r2p0</td>
</tr>
<tr>
<td>Updated configurable DFI options for clocks.</td>
<td>1.5 Configurable options on page 1-16</td>
<td>Revision r2p0</td>
</tr>
<tr>
<td></td>
<td>2.2 Clocking and resets on page 2-24</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.3.3 PHY interface on page 2-25</td>
<td></td>
</tr>
<tr>
<td>Updated information about the System interface.</td>
<td>2.1 About the functions on page 2-22</td>
<td>Revision r2p0</td>
</tr>
<tr>
<td>Added note regarding exclusive access.</td>
<td>2.3.1 System interface on page 2-25</td>
<td>Revision r2p0</td>
</tr>
<tr>
<td>Updated the DIMM support list.</td>
<td>2.4 Constraints and limitations of use on page 2-29</td>
<td>Revision r2p0</td>
</tr>
</tbody>
</table>

### Table B-5 Differences between revision 0200 issue 00 and revision 0200 issue 01

<table>
<thead>
<tr>
<th>Change</th>
<th>Location</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>There are no technical changes.</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>