

# CoreLink™ DMC-400 Dynamic Memory Controller

Revision: r0p0

## Technical Reference Manual



# CoreLink DMC-400 Dynamic Memory Controller

## Technical Reference Manual

Copyright © 2011 ARM. All rights reserved.

### Release Information

The following changes have been made to this book.

Change history			
Date	Issue	Confidentiality	Change
30 June 2011	A	Non-Confidential	First issue for revision r0p0

### Proprietary Notice

Words and logos marked with ® or ™ are registered trademarks or trademarks of ARM® in the EU and other countries, except as otherwise stated below in this proprietary notice. Other brands and names mentioned herein may be the trademarks of their respective owners.

Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder.

The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given by ARM in good faith. However, all warranties implied or expressed, including but not limited to implied warranties of merchantability, or fitness for purpose, are excluded.

This document is intended only to assist the reader in the use of the product. ARM shall not be liable for any loss or damage arising from the use of any information in this document, or any error or omission in such information, or any incorrect use of the product.

Where the term ARM is used it means “ARM or any of its subsidiaries as appropriate”.

### 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.

### Product Status

The information in this document is final, that is for a developed product.

### Web Address

<http://www.arm.com>

# Contents

## CoreLink DMC-400 Dynamic Memory Controller

### Technical Reference Manual

	<b>Preface</b>	
	About this book .....	vi
	Feedback .....	ix
<b>Chapter 1</b>	<b>Introduction</b>	
	1.1 About the AMBA DMC-400 .....	1-2
	1.2 Compliance .....	1-3
	1.3 Features .....	1-4
	1.4 Interfaces .....	1-5
	1.5 Configurable options .....	1-6
	1.6 Test features .....	1-7
	1.7 Product documentation, design flow, and architecture .....	1-8
	1.8 Product revisions .....	1-9
<b>Chapter 2</b>	<b>Functional Description</b>	
	2.1 DMC-400 interfaces .....	2-2
	2.2 DMC-400 operation .....	2-6
<b>Appendix A</b>	<b>Signal Descriptions</b>	
	A.1 ACE-Lite signals .....	A-2
	A.2 APB signals .....	A-5
	A.3 DFI signals .....	A-6
	A.4 LPI signals .....	A-10
	A.5 VN signals .....	A-11
	A.6 Ad hoc signals .....	A-12

## Appendix B      Revisions

# Preface

This preface introduces the *CoreLink DMC-400 Dynamic Memory Controller Technical Reference Manual*. It contains the following sections:

- [About this book on page vi](#)
- [Feedback on page ix.](#)

## About this book

This is the *Technical Reference Manual* (TRM) for the *CoreLink DMC-400 Dynamic Memory Controller*.

## Product revision status

The *rn**pn* identifier indicates the revision status of the product described in this book, where:

- rn** Identifies the major revision of the product.
- pn** Identifies the minor revision or modification status of the product.

## Intended audience

This book is written for system designers, system integrators, and programmers who are designing or programming a *System-on-Chip* (SoC) device that uses the DMC-400. The DMC-400 provides an interface between the *Advanced Coherency Extensions* (ACE-Lite™) system bus and external, off-chip, memory devices.

## Using this book

This book is organized into the following chapters:

### Chapter 1 Introduction

Read this for an introduction to the DMC-400 and its features.

### Chapter 2 Functional Description

Read this for an overview of the major functional blocks and the operation of the DMC-400.

### Appendix A Signal Descriptions

Read this for a description of the input and output signals.

### Appendix B Revisions

Read this for a description of 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.

The *ARM Glossary* is available on the ARM Infocenter at,  
<http://infocenter.arm.com/help/topic/com.arm.doc.aeg0014-/index.html>.

## Conventions

Conventions that this book can use are described in:

- *Typographical* on page vii
- *Timing diagrams* on page vii
- *Signals* on page vii.

## Typographical

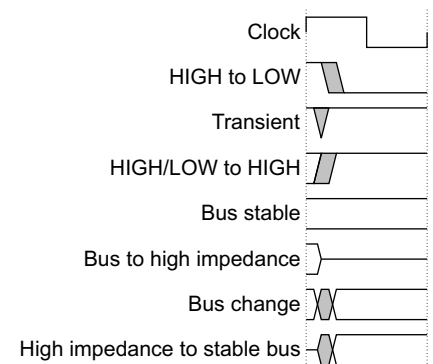
The typographical conventions are:

<i>italic</i>	Highlights important notes, introduces special terminology, denotes internal cross-references, and citations.
<b>bold</b>	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.
<u>monospace</u>	Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
<i>monospace italic</i>	Denotes arguments to monospace text where the argument is to be replaced by a specific value.
<b>monospace bold</b>	Denotes language keywords when used outside example code.
< <b>and</b> >	Enclose replaceable terms for assembler syntax where they appear in code or code fragments. For example: MRC p15, 0 <Rd>, <CRn>, <CRm>, <Opcode_2>

## Timing diagrams

The figure named *Key to timing diagram conventions* 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.



**Key to timing diagram conventions**

## Signals

The signal conventions are:

<b>Signal level</b>	The level of an asserted signal depends on whether the signal is active-HIGH or active-LOW. Asserted means: <ul style="list-style-type: none"> <li>• HIGH for active-HIGH signals</li> <li>• LOW for active-LOW signals.</li> </ul>
<b>Lower-case n</b>	At the start or end of a signal name denotes an active-LOW signal.

## Additional reading

This section lists publications by ARM and by third parties.

See Infocenter, <http://infocenter.arm.com>, for access to ARM documentation.

### ARM publications

This book contains information that is specific to this product. See the following documents for other relevant information:

- *CoreLink DMC-400 Dynamic Memory Controller Technical Reference Manual Supplement* (ARM DSU 0016)
- *CoreLink DMC-400 Dynamic Memory Controller Implementation Guide* (ARM DII 0257)
- *CoreLink DMC-400 Dynamic Memory Controller Integration Manual* (ARM DII 0258)
- *AMBA AXI Protocol Specification* (ARM IHI 0022)
- *AMBA 3 APB Protocol Specification* (ARM IHI 0024)
- *AMBA AXI and ACE Protocol Specification AXI3, AXI4, and AXI4-Lite, ACE and ACE-Lite* (ARM IHI 0022)
- *TrustZone® Address Space Controller Technical Reference Manual* (ARM DDI 0431).

### Other publications

This section lists relevant documents published by third parties:

- *DDR PHY Interface (DFI) Specification*, JEP106, <http://www.ddr-phy.org>
- *JEDEC STANDARD DDR3 SDRAM Specification*, JESD79-3D, <http://www.jedec.org>
- *JEDEC STANDARD DDR2 SDRAM Specification*, JESD79-2E, <http://www.jedec.org>
- *JEDEC STANDARD LPDDR2 SDRAM Specification*, JESD209-2 LPDDR2, <http://www.jedec.org>.



## Feedback

ARM welcomes feedback on this product and its documentation.

### 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](mailto:errata@arm.com). Give:

- the title
- the number, ARM DDI 0466A
- the page numbers to which your comments apply
- a concise explanation of your comments.

ARM also welcomes general suggestions for additions and improvements.

### Customer support

Go to *Support and Maintenance*,  
<http://www.arm.com/support/services/support-maintenance.php> and select the **Physical IP** tab for information about support contract options.

If you cannot contact ARM through the web support channel then send an e-mail to [support-pipd@arm.com](mailto:support-pipd@arm.com).

---

**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 introduces the DMC-400 and contains the following sections:

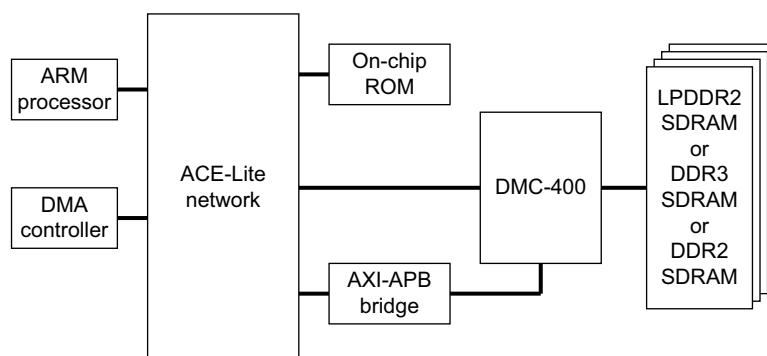
- *About the AMBA DMC-400* on page 1-2
- *Compliance* on page 1-3
- *Features* on page 1-4
- *Interfaces* on page 1-5
- *Configurable options* on page 1-6
- *Test features* on page 1-7
- *Product documentation, design flow, and architecture* on page 1-8
- *Product revisions* on page 1-9.

## 1.1 About the AMBA DMC-400

The DMC-400 is an *Advanced Microcontroller Bus Architecture* (AMBA) compliant *System-on-Chip* (SoC) peripheral that is developed, tested, and licensed by ARM.

It is a high-performance, area-optimized, *Low Power Double Data Rate 2* (LPDDR2) *Synchronous Dynamic Random Access Memory* (SDRAM), *Double Data Rate 3* (DDR3) SDRAM, and *Double Data Rate 2* (DDR2) SDRAM memory controller that is compatible with the AMBA ACE-Lite protocol.

Figure 1-1 shows an example system.



**Figure 1-1 Example system**

The DMC-400 enables data transfer between the SoC and the DRAM devices external to the chip. It connects to the on-chip system through one or more ACE-Lite interfaces. It connects to the DRAM devices through its memory interface block, and the DFI pad interface.

## 1.2 Compliance

The DMC-400 is compliant with the following standards and protocols:

- ACE-Lite
- AMBA3 APB protocol
- JEDEC LPDDR2 JESD209-2 standard
- JEDEC DDR3 JESD79-3D standard
- JEDEC DDR2 JESD79-2E standard
- DFI.

## 1.3 Features

The DMC-400 features are:

- soft macrocell available in Verilog
- configurable hardware to support area and performance optimization
- supports multiple ACE-Lite system interfaces
- supports multiple outstanding transactions
- supports system *Quality of Service* (QoS) and request arbitration features, to achieve low latency transfers and the optimal use of memory bandwidth
- protects against transaction barriers
- configurable bit width for the ACE-Lite interfaces
- separate read acceptance capability and write acceptance capability
- supports ARMv6 architecture for exclusive access transfers
- synchronous n:1 clocking between ACE-Lite and APB
- supports LPDDR2-S2, LPDDR2-S4, DDR3, and DDR2 memory devices
- supports multiple memory interfaces
- optimal use of the external memory bus
- comprehensive support for DRAM power saving features
- programmable external memory width
- supports a configurable number of memory chip selects for each memory interface
- supports memory *Error-Correcting Code* (ECC)
- interfaces to the PHY of the DRAM devices using a DFI pad interface
- supports the DFI2.1 specification of the *Physical Layer* (PHY) interface.

## 1.4 Interfaces

The DMC-400 has the following external interfaces:

- APB3 interface, to configure and control the DMC-400
- ACE-Lite slave interface, to transfer memory data to or from an AMBA master
- DFI-compatible PHY interface, to transfer data to or from the memory devices
- clock and reset interface
- debug and profile interface
- low-power control interface
- training interface.

See [DMC-400 interfaces on page 2-2](#) for more information.

## 1.5 Configurable options

The DMC-400 has the following configurable options:

- number of ACE-Lite system interfaces
- for all ACE-Lite system interfaces:
  - data bus width
  - address bus width
  - ID bus width
  - read acceptance capability
  - read hazard acceptance capability.
- for all memory interfaces:
  - read queue depth
  - write buffer depth.
- number of memory channels
- number of memory chips for all memory channels
- effective data width for all memory channels
- maximum effective memory burst length
- enable or disable memory *Single-Error Correction and Double-Error Detection* (SECEDED).

See [Configurability on page 2-8](#) for more information.

## 1.6 Test features

The DMC-400 provides:

- integration test logic for integration testing
- the debug and profile interface to enable you to monitor transaction events.



## 1.7 Product documentation, design flow, and architecture

The DMC-400 documentation is as follows:

### Technical Reference Manual and Technical Reference Manual Supplement

The *Technical Reference Manual* (TRM) and TRM Supplement describe the functionality and the effects of functional options on the behavior of the DMC-400. These are required at all stages of the design flow. The choices that you make in the design flow can mean that some behavior that the TRM describes is not relevant. If you are programming the DMC-400 then contact:

- the implementer to determine:
  - the build configuration of the implementation
  - what integration, if any, was performed before implementing the DMC-400.
- the integrator to determine the pin configuration of the device that you are using.

The TRM Supplement is a confidential book that is only available to licensees.

### Implementation Guide

The *Implementation Guide* (IG) describes:

- the available build configuration options and related issues in selecting them
- how to configure the *Register Transfer Level* (RTL) with the build configuration options
- the processes to sign off the configured design.

The ARM product deliverables include reference scripts and information about using them to implement your design.

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

### Integration Manual

The *Integration Manual* (IM) describes how to integrate the DMC-400 into a SoC. It includes a description of the pins that the integrator must tie off to configure the macrocell for the required integration. Some of the integration is affected by the configuration options used when implementing the DMC-400.

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

## 1.8 Product revisions

This section describes the differences in functionality between product revisions of the DMC-400:

**r0p0** First release.

# Chapter 2

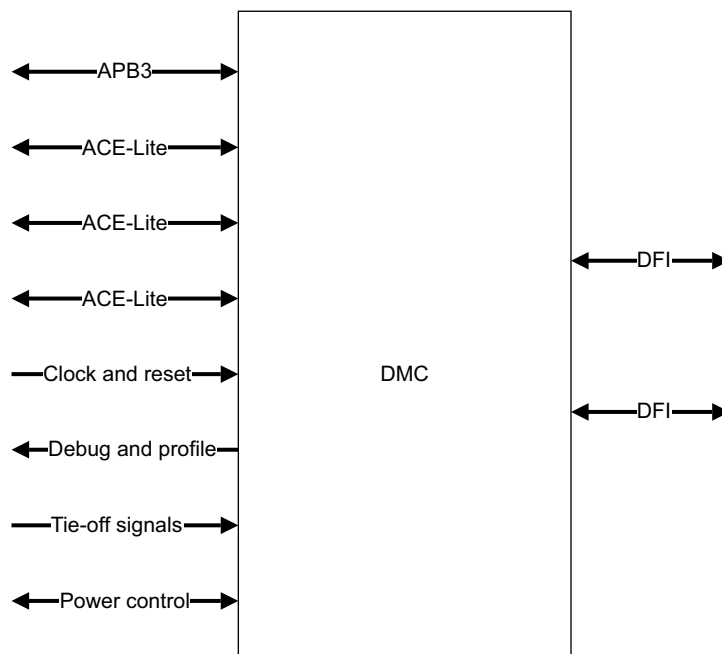
## Functional Description

This chapter describes the DMC-400 operation. It contains the following sections:

- [DMC-400 interfaces on page 2-2](#)
- [DMC-400 operation on page 2-6.](#)

## 2.1 DMC-400 interfaces

Figure 2-1 shows the interfaces of the DMC-400.



**Figure 2-1 DMC-400 interfaces**

The interfaces of the DMC-400 are:

- [APB3 interface](#)
- [ACE-Lite interface on page 2-3](#)
- [PHY, DFI 2.1, interface on page 2-3](#)
- [Clocks and resets on page 2-4](#)
- [Debug and profile interface on page 2-5](#)
- [Hardware power control interface\(s\) on page 2-5](#)
- [Training interface on page 2-5.](#)

### 2.1.1 APB3 interface

You can access the configuration and control registers of the DMC-400 through the APB3 interface. Use these registers for initialization, configuration, control, and power-management of the DMC-400.

The APB3 interface address map has a range of 4KB with which you can access the DMC-400 control registers. The address map is split into regions. [Table 2-1](#) shows the function of each region.

**Table 2-1 APB3 register regions**

Region base address	Region function
0x000	Memory controller configuration
0x100	Memory channel configuration
0x200	PHY control

Table 2-1 APB3 register regions (continued)

Region base address	Region function
0x300	Timing configuration
0x400	User configuration
0xE00	Integration test
0xF00	Peripheral ID

See the *AMBA 3 APB Protocol Specification* for more information on the APB3 interface.

See [Appendix A Signal Descriptions](#) for more information on the APB3 signals that the DMC-400 uses.

---

**Note**

The DMC-400 does not produce PLSVERR responses. Incorrect use of the APB interface causes either a failure without notification, or undefined behavior.

---

### 2.1.2 ACE-Lite interface

The DMC-400 uses the standard ACE-Lite interface as the *AMBA AXI and ACE Protocol Specification AXI3, AXI4, and AXI4-Lite, ACE and ACE-Lite* describes. See [Appendix A Signal Descriptions](#) for a description of the interface signals.

For DMC-400 configurations that have more than one system interface, the signal names for each system interface include the suffix `<_index>`, where `<_index>` is the decimal system address.

---

**Note**

The DMC-400 does not support the **axprot** signal.

---

The following signals have configurable widths:

- **axaddr**
- **axid**
- **xdata**
- **wstrb**.

where x is either w for write, or r for read.

### 2.1.3 PHY, DFI 2.1, interface

The PHY interface connects the DMC-400 to the memory devices. This interface conforms to the DFI2.1 interface standard. See the *DDR PHY Interface Specification JEP106*.

For DMC-400 configurations that have more than one memory interface, the signal names for each memory interface include the suffix `<_index>`, where `<_index>` is the decimal channel address.

The following signals have configurable widths:

- **dfi\_cke**
- **dfi\_dram\_clk\_disable**
- **dfi\_odt**

- **dfi\_rddata**
- **dfi\_rddata\_en**
- **dfi\_wrdata**
- **dfi\_wrdata\_en**
- **dfi\_wrdata\_mask**.

---

**Note**

---

The DMC-400 does not support the following DFI signals:

- **dfi\_ctrlupd\_req**
  - **dfi\_ctrlupd\_ack**.
- 

The DMC-400 supports the Update Request handshake that the DFI PHY initiates. The PHY, or an agent working on behalf of the PHY, can request an update through the DFI interface by asserting **dfi\_phyupd\_req**.

## 2.1.4 Clocks and resets

Table 2-2 shows the clock and reset signals of the DMC-400 when you configure it to have the maximum number of system interfaces, and maximum number of memory interfaces.

**Table 2-2 Clock and reset signals**

Signal name	Width	Direction	Description
<b>dmc_clk</b>	1	Input	Clock for the DMC
<b>pclk</b>	1	Input	Clock for the APB interface clock domain
<b>dmc_resetsn</b>	1	Input	Reset for the DMC
<b>presetsn</b>	1	Input	Reset for APB interface domain

### Clocks

The DMC clock and the APB interface clock must be phase-aligned, synchronous. The DMC clock must be of a frequency higher than, or equal to the frequency of the APB clock.

---

**Note**

---

You must only change the frequency of the DMC clock when both of the following conditions are true:

- all memory chips are in self-refresh mode or in *Deep Power Down* (DPD) mode
  - the DMC-400 status registers indicate that the memory interface is in the quiescent state.
- 

### Reset

Use the **dmc\_resetsn** signal to reset the DMC-400. You must assert the **dmc\_resetsn** signal for longer than two **dmc\_clk** clock periods. You can assert **dmc\_resetsn** asynchronously to **sx\_clk**. You must de-assert **dmc\_resetsn** synchronously with the rising edge of **dmc\_clk**.

Use the **presetsn** signal to reset the APB interface of the DMC-400. You must assert the **presetsn** signal for longer than two **pclk** clock periods. You can assert **presetsn** asynchronously to **pclk**. You must de-assert **presetsn** synchronously with the rising edge of **pclk**.

**Note**

To assert any DMC-400 reset signal you must set it LOW.

### 2.1.5 Debug and profile interface

The debug and profile interface enables you to monitor events that occur within the DMC. You can use these events to debug and to monitor the performance of the DMC.

### 2.1.6 Hardware power control interface(s)

The DMC-400 supports the standard AXI Low-power Interface. See the *AMBA 3 AXI Protocol Specification*.

For DMC-400 configurations that have more than one memory interface, the interface signal names include the suffix `<_index>`, where `<_index>` is the decimal channel address.

### 2.1.7 Training interface

The DFI specification defines evaluation modes for all of the following:

- write leveling training
- read data eye training
- gate training.

Table 2-3 shows the evaluation modes that the DMC-400 hardware supports.

**Table 2-3 DFI evaluation modes**

Mode	Description
No training in the PHY	-
PHY evaluation	The DMC-400 controls the training logic. The PHY determines the correct delay values.
PHY independent	The PHY controls all training operations.
<i>Memory Controller (MC) evaluation</i>	The DMC-400 does not support this evaluation mode.

## 2.2 DMC-400 operation

This section describes:

- [Memory initialization](#)
- [Memory support](#)
- [Memory access](#)
- [Power control on page 2-7](#)
- [Configurability on page 2-8](#)
- [Security on page 2-8](#).

### 2.2.1 Memory initialization

You can initialize the DRAM memory devices through the APB3 interface.

### 2.2.2 Memory support

The DMC-400 supports multiple memory channels. Each channel can support the following memory types:

- LPDDR2-S2
- LPDDR2-S4
- DDR3
- DDR2
- low-voltage DDR3.

The DMC-400 is a configurable peripheral. It enables you to:

- manage the distribution of traffic across a number of memory interfaces
- minimize the overhead, and cyclic dependency between channels
- use its flexibility to create an efficient memory subsystem
- manage the operation and maintenance of multi-port memory devices.

You can use each chip-select signal to select a group of devices, that are bound, to form a single data interface.

### 2.2.3 Memory access

The DMC-400 connects to the system through one or more ACE-Lite interfaces. The DMC-400 supports all the features of the ACE-Lite specification. The features include:

- barriers
- cache maintenance operations
- QoS.

#### Addressing

The DMC-400 maps the ACE-Lite address to a physical DRAM address. The DRAM address consists of a bank number, a row number, and a column number. The DMC-400 supports configurable and programmable controls for mapping the ACE-Lite address to the physical DRAM address.

#### ———— Note ————

DRAM devices have page sizes of 1, 2, or 4KB. The ACE-Lite protocol does not permit a transaction to cross a 4KB boundary.



The DMC-400 provides programmable controls to map the system address onto the physical chips and channels.

### **Burst control**

The DMC-400 formats all ACE-Lite transactions into memory bursts.

The DMC-400 supports sequential burst addressing, and sequential wrapped burst addressing of the DRAM.

### **Responses**

The DMC-400 always replies to an ACE-Lite access with either an OKAY response or an EXOKAY response. It never issues a SLVERR response or a DECERR response.

### **Exclusive access**

The DMC-400 provides eight exclusive access monitors for each memory interface.

### **Barriers and cache maintenance operations**

The DMC-400 supports barriers, and cache maintenance operations to guarantee the correct ordering of memory access.

### **QoS signals**

The DMC-400 uses the **arqos** and **awqos** signals, of the ACE-Lite address, to determine the quality of service value that the system requests for its memory access.

### **Error-correcting code**

The DMC-400 provides SECDED code protection for DRAM accesses. It combines the codes with data for write transactions, and checks the codes for data read transactions.

The DMC-400 uses a programmable interrupt to report errors that it corrects. You can read the information of a transaction that the DMC-400 corrects through an APB register in the DMC-400. You can clear the interrupt through an APB register.

The DMC-400 reports uncorrected errors through another programmable interrupt. You can clear the interrupt through an APB register. You can read the information of the transaction through an APB register.

If either interrupt overflows the interrupt system, the DMC-400 triggers another interrupt. You can clear this interrupt through an APB register.

## **2.2.4 Power control**

The DMC-400 and DRAM have several operating states. Each operating state consumes different amounts of power. The DMC-400 enables you to control its operating state and the operating state of the DRAM to reduce the power consumption of the DMC-400, the PHY, the interface, and the DRAMs.

### **Hardware-controlled power-management**

The DMC-400 provides hardware-controlled power-management of each memory interface through the PHY low-power request interface. This interface enables power-management hardware to remove power from idle memory interfaces and associated logic.

## Clock frequency adjustment

You can change the DMC-400 clock frequency.

### Warning

You must only do this when:

1. All memory chips are either in self-refresh mode or DPD mode.
2. The status registers indicate that the memory interface is in the quiescent state.

## DMC-400 controlled power-management

The DMC-400 supports several features to manage the power dissipation of each memory device in the memory subsystem. These programmable features enable the DMC-400 to automatically manage the entry and exit of the DRAM low-power states when the DRAM is idle.

### 2.2.5 Configurability

Table 2-4 shows the configurable parameters that you can set at design-time.

**Table 2-4 Configurable design-time parameters**

Parameter	Permitted values	Default
Configurable system interface widths	64-, 128-, 256-bit	64-bit
Configurable number of system interface address bits	32, 40, 64	32
Configurable number of system interface ID widths	4-24	8
Read acceptance capability <sup>a</sup> for system interface	16, 32, 64	32
Read hazard acceptance <sup>a</sup> for memory interface	8, 16	8
Read queue depth <sup>a</sup> for memory interface	16, 32, 64	32
Write buffer depth <sup>a</sup> for memory interface	16, 32, 64	16
Number of memory chips per memory interface	1, 2	1
Effective number of bits in the memory interface	32, 64, 128	32
Maximum effective memory burst length <sup>b</sup>	4, 8	8
SECCDED ECC support	Y N	N

a. Number of memory bursts.

b. Determines the write buffer entry size.

You must select the following configurable parameters by choosing the correct Verilog file:

- number of system interfaces
- number of memory channels.

### 2.2.6 Security

The DMC-400 does not offer any security control.

---

**Note**

---

- to mark regions of either the ACE-Lite, or APB address maps as secure, ARM recommends that you use the *TrustZone Address Space Controller* (TZASC)
  - if you mark any memory region as secure, you must also mark the APB interface as secure.
-

# Appendix A

## Signal Descriptions

This appendix describes the signals that the DMC-400 provides. It contains the following sections:

- *ACE-Lite signals* on page A-2
- *APB signals* on page A-5
- *DFI signals* on page A-6
- *LPI signals* on page A-10
- *VN signals* on page A-11
- *Ad hoc signals* on page A-12.

## A.1 ACE-Lite signals

Table A-1 shows the ACE-Lite signals.

**Table A-1 ACE-Lite interface signals**

Signal	Type	Description
<b>awid</b>	Output	Write address ID. This signal is the identification tag for the write address group of signals.
<b>awaddr</b>	Output	Write address. The write address bus gives the address of the first transfer in a write burst transaction. The associated control signals are used to determine the addresses of the remaining transfers in the burst.
<b>awregion</b>	Output	Region identifier. Enables a single physical interface on a slave to be used for multiple logical interfaces.
<b>awdomain</b>	Output	This signal indicates the shareability domain of a write transaction.
<b>awsnoop</b>	Output	This signal indicates the transaction type for shareable write transactions.
<b>awbar</b>	Output	This signal indicates a write barrier transaction.
<b>awlen</b>	Output	Burst length. The burst length gives the number of transfers in a burst. This information determines the number of data transfers associated with the address.
<b>awsize</b>	Output	Burst size. This signal indicates the size of each transfer in the burst. Byte lane strobes indicate which byte lanes to update.
<b>awburst</b>	Output	Burst type. The burst type, coupled with the size information, details how the address for each transfer within the burst is calculated.
<b>awlock</b>	Output	Lock type. This signal provides additional information about the atomic characteristics of the transfer.
<b>awcache</b>	Output	Cache type. This signal indicates the bufferable, cacheable, write-through, write-back, and allocate attributes of the transaction.
<b>awprot</b>	Output	Protection type. This signal indicates the normal, privileged, or secure protection level of the transaction and whether the transaction is a data access or an instruction access.
<b>awqos</b>	Output	Quality of Service. This signal is used to provide a QoS identifier for each write transaction.
<b>awvalid</b>	Output	Write address valid. This signal indicates that valid write address and control information are available: <b>1</b> Address and control information available. <b>0</b> Address and control information not available. The address and control information remain stable until the address acknowledge signal, <b>awready</b> , goes HIGH.
<b>awready</b>	Input	Write address ready. This signal indicates that the slave is ready to accept an address and associated control signals: <b>1</b> Slave ready. <b>0</b> Slave not ready.
<b>wdata</b>	Output	Write data. The write data bus can be 8-bit, 16-bit, 32-bit, 64-bit, 128-bit, 256-bit, 512-bit, or 1024 bit wide.
<b>wstrb</b>	Output	Write strobes. This signal indicates which byte lanes to update in memory. There is one write strobe for each eight bits of the write data bus.
<b>wlast</b>	Output	Write last. This signal indicates the last transfer in a write burst.
<b>wvalid</b>	Output	Write valid. This signal indicates that valid write data and strobes are available: <b>1</b> Write data and strobes available. <b>0</b> Write data and strobes not available.

Table A-1 ACE-Lite interface signals (continued)

Signal	Type	Description
<b>wready</b>	Input	Write ready. This signal indicates that the slave can accept the write data: <b>1</b> Slave ready. <b>0</b> Slave not ready.
<b>bid</b>	Input	Response ID. The identification tag of the write response. The <b>bid</b> value must match the <b>awid</b> value of the write transaction to which the slave is responding.
<b>bresp</b>	Input	Write response. This signal indicates the status of the write transaction. The permitted responses are OKAY, EXOKAY, SLVERR, and DECERR.
<b>bvalid</b>	Input	Write response valid. This signal indicates that a valid write response is available: <b>1</b> Write response available. <b>0</b> Write response not available.
<b>bready</b>	Output	Response ready. This signal indicates that the master can accept the response information: <b>1</b> Master ready. <b>0</b> Master not ready.
<b>arid</b>	Output	Read address ID. This signal is the identification tag for the read address group of signals.
<b>araddr</b>	Output	Read address. The read address bus gives the initial address of a read burst transaction. Only the start address of the burst is provided and the control signals that are issued alongside the address detail how the address is calculated for the remaining transfers in the burst.
<b>arregion</b>	Output	Region decode signal. The signal is provided alongside the transaction address. The signal enables a single physical interface on a slave to be used for multiple logical interfaces which reside in different locations in the system address map.
<b>ardomain</b>	Output	This signal indicates the shareability domain of a read transaction.
<b>arsnoop</b>	Output	This signal indicates the transaction type for shareable read transactions.
<b>arbar</b>	Output	This signal indicates a read barrier transaction.
<b>arlen</b>	Output	Burst length. The burst length gives the exact number of transfers in a burst. This information determines the number of data transfers associated with the address.
<b>arsize</b>	Output	Burst size. This signal indicates the size of each transfer in the burst.
<b>arburst</b>	Output	Burst type. The burst type, coupled with the size information, details how the address for each transfer within the burst is calculated.
<b>arlock</b>	Output	Lock type. This signal provides additional information about the atomic characteristics of the transfer.
<b>arcache</b>	Output	Cache type. This signal provides additional information about the cacheable characteristics of the transfer.
<b>arprot</b>	Output	Protection type. This signal provides protection unit information for the transaction.
<b>arqos</b>	Output	Quality of Service. Signal used to provide a QoS identifier for each read transaction.
<b>arvalid</b>	Output	Read address valid. When HIGH this signal indicates that the read address and control information is valid and stable until the <b>arready</b> address acknowledge signal is HIGH: <b>1</b> Address and control information valid. <b>0</b> Address and control information not valid.
<b>arready</b>	Input	Read address ready. This signal indicates that the slave is ready to accept an address and associated control signals: <b>1</b> Slave ready. <b>0</b> Slave not ready.

Table A-1 ACE-Lite interface signals (continued)

Signal	Type	Description
<b>rid</b>	Input	Read ID tag. This signal is the ID tag of the read data group of signals. The <b>rid</b> value is generated by the slave and must match the <b>arid</b> value of the read transaction to which it is responding.
<b>rdata</b>	Input	Read data. The read data bus can be 8-bit, 16-bit, 32-bit, 64-bit, 128-bit, 256-bit, 512-bit, or 1024-bit wide.
<b>rresp</b>	Input	Read response. This signal indicates the status of the read transfer. The permitted responses are OKAY, EXOKAY, SLVERR, and DECERR.
<b>rlast</b>	Input	Read last. This signal indicates the last transfer in a read burst.
<b>rvalid</b>	Input	Read valid. This signal indicates that the required read data is available and the read transfer can complete: <b>1</b> Read data available. <b>0</b> Read data not available.
<b>rready</b>	Output	Read ready. This signal indicates that the master can accept the read data and response information: <b>1</b> Master ready. <b>0</b> Master not ready.
<b>awuser</b>	Output	Optional user-defined signal in the write address channel.
<b>wuser</b>	Output	Optional user-defined signal in the write data channel.
<b>buser</b>	Input	Optional user-defined signal in the write response channel.
<b>aruser</b>	Output	Optional user-defined signal in the read address channel.
<b>ruser</b>	Input	Optional user-defined signal in the read data channel.

## A.2 APB signals

Table A-2 shows the APB signals.

**Table A-2 APB interface signals**

Signal	Type	Description
<b>pclk</b>	Input	Clock. The rising edge of <b>pclk</b> synchronizes all transfers on the APB.
<b>presetn</b>	Input	Reset. The APB reset signal is active LOW. This signal is normally connected directly to the system bus reset signal.
<b>paddr</b>	Output	Address. This is the APB address bus. It can be up to 32-bits wide and is driven by the peripheral bus bridge unit.
<b>psel</b>	Output	Select. The APB bridge unit generates this signal to each peripheral bus slave. It indicates that the slave device is selected and that a data transfer is required. There is a <b>psel</b> signal for each slave.
<b>penable</b>	Output	Enable. This signal indicates the second and subsequent cycles of an APB transfer.
<b>pwrite</b>	Output	Direction. This signal indicates an APB write access when HIGH and an APB read access when LOW.
<b>prdata</b>	Input	Read Data. The selected slave drives this bus during read cycles when <b>pwrite</b> is LOW. This bus can be up to 32-bits wide.
<b>pwdata</b>	Output	Write data. This bus is driven by the peripheral bus bridge unit during write cycles when <b>pwrite</b> is HIGH. This bus can be up to 32-bits wide.
<b>pready</b>	Input	Ready. The slave uses this signal to extend an APB transfer.
<b>pslverr</b>	Input	This signal indicates a transfer failure. APB peripherals are not required to support the <b>pslverr</b> pin. This is true for both existing and new APB peripheral designs. When a peripheral does not include this pin, the appropriate input to the APB bridge is tied LOW.
<b>pclken</b>	Input	This is an optional enable signal for <b>pclk</b> domain.



## A.3 DFI signals

Table A-3 shows the DFI signals.

**Table A-3 DFI interface signals**

Signal	Type	Description
<b>dfi_init_complete</b>	Output	PHY initialization complete. The <b>dfi_init_complete</b> signal indicates that the PHY is able to respond to any proper stimulus on the DFI. All DFI signals that communicate commands or status must be held at their default values until the <b>dfi_init_complete</b> signal asserts. During a PHY re-initialization request, for example a frequency change, this signal is de-asserted. For a frequency change request, the de-assertion of the <b>dfi_init_complete</b> signal acknowledges the frequency change protocol. Once de-asserted, the signal should only be re-asserted within <b>tinit_complete</b> cycles after the <b>dfi_init_start</b> signal has deasserted, and after the PHY has completed re-initialization.
<b>dfi_rddata_valid</b>	Input	Read data valid indicator. The <b>dfi_rddata_valid</b> signal is asserted with the read data for the number of cycles that data is being sent. The timing is the same as for the <b>dfi_rddata</b> bus.
<b>dfi_rddata</b>	Input	Read data bus. Read data is expected to be received at the MC within <b>tphy_rdlat</b> cycles after the <b>dfi_rddata_en</b> signal is asserted.
<b>dfi_rdecc</b>	Input	Read data ECC code bus. The timing is the same as for the <b>dfi_rddata</b> bus.
<b>dfi_rddata_en</b>	Output	Read data enable. The <b>dfi_rddata_en</b> signal must be asserted <b>trddata_en</b> cycles after the assertion of a read command on the DFI control interface and remains valid for the duration of contiguous read data expected on the <b>dfi_rddata</b> bus.
<b>dfi_rdecc_en</b>	Output	Read data ECC code enable. The timing is the same as for the <b>dfi_rddata_en</b> bus.
<b>dfi_ras_n</b>	Output	DFI row address strobe bus. These signals define the RAS information that is intended for the DRAM devices for all control commands.
<b>dfi_cas_n</b>	Output	DFI column address strobe bus. These signals define the CAS information that is intended for the DRAM devices for all control commands.
<b>dfi_we_n</b>	Output	DFI write enable bus. These signals define the WEN information that is intended for the DRAM devices for all control commands.
<b>dfi_cke</b>	Output	DFI clock enable bus. These signals define the CKE information that is intended for the DRAM devices for all control commands.
<b>dfi_cs_n</b>	Output	DFI chip select bus. These signals define the chip select information that is intended for the DRAM devices for all control commands.
<b>dfi_cmd_addr</b>	Output	DFI command/address bus. These signals define the CA information that is intended for the DRAM memory devices with muxed CA signalling, for example LPDDR2.
<b>dfi_address</b>	Output	DFI address bus. These signals define the address information that is intended for the DRAM memory devices for all control commands.
<b>dfi_bank</b>	Output	DFI bank bus. These signals define the bank information that is intended for the DRAM devices for all control commands.
<b>dfi_odt</b>	Output	DFI on-die termination control bus. These signals define the ODT information that is intended for the DRAM devices for all control commands.
<b>dfi_wrdata</b>	Output	Write data bus. The write data stream must begin <b>tphy_wrdata</b> cycles after the <b>dfi_wrdata_en</b> signal is asserted for the number of cycles that the <b>dfi_wrdata_en</b> signal is asserted. If the PHY requires notification of pending write data sooner, the <b>tphy_wrdata</b> timing parameter might be adjusted to a higher value.

Table A-3 DFI interface signals (continued)

Signal	Type	Description
<b>dfi_wrdata_mask</b>	Output	Write-data byte mask. The timing is the same as for the <b>dfi_wrdata</b> bus. For example, the <b>dfi_wrdata_mask [0]</b> signal defines masking for the <b>dfi_wrdata [7:0]</b> signals, the <b>dfi_wrdata_mask [1]</b> signal defines masking for the <b>dfi_wrdata [15:8]</b> signals. If the <b>dfi_wrdata</b> bus is not a multiple of 8 bits, then the uppermost bit of the <b>dfi_wrdata_mask</b> signal corresponds to the most significant partial byte of data.
<b>dfi_wrdata_en</b>	Output	Write data and data mask valid. These signals must be asserted <b>tphy_wrdata</b> cycles before the data and data mask are sent on the DFI interface. If the PHY requires notification of pending write data sooner, the <b>tphy_wrdata</b> timing parameter may be adjusted to a higher value. The <b>dfi_wrdata_en</b> signal must be sent <b>tphy_wrlat</b> cycles after the write command. When the <b>dfi_wrdata_en</b> signal is asserted, it must remain asserted for the number of contiguous cycles of write-data passed through the DFI write-data interface. The width of the <b>dfi_wrdata_en</b> signal is defined as a DFI term. There must be a single <b>dfi_wrdata_en</b> bit for each slice of memory data.
<b>dfi_wrecc</b>	Output	Write data ECC code bus. Has the same timing as the <b>dfi_wrdata</b> signal.
<b>dfi_wrecc_mask</b>	Output	Write data ECC code byte mask. Has the same timing as the <b>dfi_wrdata_mask</b> signal.
<b>dfi_wrecc_en</b>	Output	Write data ECC code and mask valid. It has the same timing as the <b>dfi_wrdata_en</b> signal.
<b>dfi_dram_clk_disable</b>	Output	DRAM clock disable. When active, this indicates to the PHY that the clocks to the DRAM devices must be disabled so that the clock signals hold a constant value. When the <b>dfi_dram_clk_disable</b> signal is inactive, the DRAMs must be clocked normally.
<b>dfi_phyupd_req</b>	Input	PHY-initiated update request. The <b>dfi_phyupd_req</b> signal is used for a PHY-initiated update to indicate that the PHY requires the DFI to not send control, read or write commands or data for a specified period of time. When asserted, the <b>dfi_phyupd_req</b> signal must remain asserted until the request is acknowledged by the assertion of the <b>dfi_phyupd_ack</b> signal and the update has been completed. The MC must acknowledge this request.
<b>dfi_phyupd_ack</b>	Output	PHY-initiated update acknowledge. The <b>dfi_phyupd_ack</b> signal is used for a PHY-initiated update to indicate that the DFI is idle and remains so until the <b>dfi_phyupd_req</b> signal de-asserts. While this signal is asserted, the DFI bus must remain idle other than any transactions specifically associated with the update process.
<b>dfi_phyupd_type</b>	Input	PHY-initiated update select. The <b>dfi_phyupd_type</b> signal indicates which one of the 4 types of PHY update times is being requested by the <b>dfi_phyupd_req</b> signal.
<b>dfi_clp_req</b>	Output	Low-power opportunity request. The <b>dfi_lp_req</b> signal is used by the MC to inform the PHY of an opportunity to switch to a low-power mode. This signal corresponds to command signals and associated logic.
<b>dfi_clp_ack</b>	Input	Low-power acknowledge. The <b>dfi_lp_ack</b> signal is asserted to acknowledge the MC low-power opportunity request. The PHY is not required to acknowledge this request. This signal corresponds to command signals and associated logic.
<b>dfi_clp_wakeup</b>	Output	Low-power wakeup time. The <b>dfi_lp_wakeup</b> signal indicates which one of the 16 wakeup times the MC is requesting for the PHY. This signal corresponds to command signals and associated logic.
<b>dfi_rdlp_req</b>	Output	Low-power opportunity request. The <b>dfi_lp_req</b> signal is used by the MC to inform the PHY of an opportunity to switch to a low-power mode. This signal corresponds to read data signals and associated logic.
<b>dfi_rdlp_ack</b>	Input	Low-power acknowledge. The <b>dfi_lp_ack</b> signal is asserted to acknowledge the MC low-power opportunity request. The PHY is not required to acknowledge this request. This signal corresponds to read data signals and associated logic.

Table A-3 DFI interface signals (continued)

Signal	Type	Description
<b>dfi_rdlp_wakeup</b>	Output	Low-power wakeup time. The <b>dfi_lp_wakeup</b> signal indicates which one of the 16 wakeup times the MC is requesting for the PHY. This signal corresponds to read data signals and associated logic.
<b>dfi_wrlp_req</b>	Output	Low-power opportunity request. The <b>dfi_lp_req</b> signal is used by the MC to inform the PHY of an opportunity to switch to a low-power mode. This signal corresponds to write data signals and associated logic.
<b>dfi_wrlp_ack</b>	Input	Low-power acknowledge. The <b>dfi_lp_ack</b> signal is asserted to acknowledge the MC low-power opportunity request. The PHY is not required to acknowledge this request. This signal corresponds to write data signals and associated logic.
<b>dfi_wrlp_wakeup</b>	Output	Low-power wakeup time. The <b>dfi_lp_wakeup</b> signal indicates which one of the 16 wakeup times the MC is requesting for the PHY. This signal corresponds to write data signals and associated logic.
<b>dfi_rdlvl_req</b>	Input	PHY-initiated read data eye training request.
<b>dfi_rdlvl_en</b>	Output	PHY data eye training logic enable. If the PHY initiates the training request, see the <b>dfi_rdlvl_req</b> signal, then this serves as an acknowledge of that request: <b>1</b> Training logic enabled. <b>0</b> Normal operation. The assertion of this signal immediately triggers read training.
<b>dfi_rdlvl_gate_req</b>	Input	PHY-initiated read gate training request.
<b>dfi_rdlvl_gate_en</b>	Output	PHY gate training logic enable. If the PHY initiates the training request, see the <b>dfi_rdlvl_gate_req</b> signal, then this serves as an acknowledge of that request: <b>1</b> Training logic enabled. <b>0</b> Normal operation. The assertion of this signal immediately triggers read training.
<b>dfi_rdlvl_resp</b>	Input	Read leveling response. Response definition depends on the mode of operation and the memory type for the system: <b>PHY Evaluation mode for DDR3 memory systems</b> The response indicates that the PHY has completed read leveling and centered the DQS relative to the data or placed the gate within the DQS preamble. <b>PHY Evaluation mode for LPDDR2 memory systems</b> The response indicates that the PHY has completed data eye training or gate training and centered the DQS relative to the data or placed the gate within the DQS preamble.
<b>dfi_wrlvl_req</b>	Input	PHY-initiated write training request.

Table A-3 DFI interface signals (continued)

Signal	Type	Description
<b>dfi_wrlvl_en</b>	Output	<p>PHY write training logic enable. If the PHY initiates the training request, see the <b>dfi_wrlvl_req</b> signal, then this serves as an acknowledge of that request:</p> <p><b>1</b> Training logic enabled.</p> <p><b>0</b> Normal operation.</p> <p>The assertion of this signal immediately triggers data eye training.</p>
<b>dfi_wrlvl_strobe</b>	Output	Write leveling strobe. Triggers the PHY write leveling strobe.
<b>dfi_wrlvl_resp</b>	Input	<p>Write leveling response. Response definition depends on the mode of operation:</p> <p><b>PHY Evaluation mode</b></p> <p>The response indicates that the PHY has completed write leveling and aligned the DQS relative to the memory clock.</p> <p><b>MC Evaluation mode</b></p> <p>The response indicates the sampled level of DQ.</p> <p>This value is used by the MC to determine how to adjust the delay value.</p>

## A.4 LPI signals

Table A-4 shows the LPI signals.

**Table A-4 LPI interface signals**

Signal	Type	Description
<b>csysreq</b>	Output	System low-power request. This signal is a request from the system clock controller for the peripheral to enter a low-power state.
<b>csysack</b>	Input	Low-power request acknowledgement. This signal is the acknowledgement from a peripheral of a system low-power request.
<b>cactive</b>	Input	Clock active. This signal indicates that the peripheral requires its clock signal: <b>1</b> Peripheral clock required. <b>0</b> Peripheral clock not required.

## A.5 VN signals

Table A-5 shows the VN signals.

**Table A-5 VN interface signals**

Signal	Type	Description
<b>vawvalid</b>	Output	Token request valid. This signal indicates that a token is being requested.
<b>vawready</b>	Input	Token request accepted. This signal indicates that the slave is ready to accept an address and associated control signals: <b>1</b> Slave ready. <b>0</b> Slave not ready.
<b>vawqos</b>	Output	Quality of service value. Non-standard AXI3 signal.
<b>vwvalid</b>	Output	Token request valid. This signal indicates that a token is being requested.
<b>vwready</b>	Input	Token request accepted. This signal indicates that the slave is ready to accept an address and associated control signals: <b>1</b> Slave ready. <b>0</b> Slave not ready.
<b>varvalid</b>	Output	Token request valid. This signal indicates that a token is being requested:
<b>varready</b>	Input	Token request accepted. This signal indicates that the slave is ready to accept an address and associated control signals: <b>1</b> Slave ready. <b>0</b> Slave not ready.
<b>varqos</b>	Output	Quality of service value. Non standard AXI3 signal.

## A.6 Ad hoc signals

Table A-6 shows the ad hoc signals.

**Table A-6 Ad hoc signals**

Signal	Type	Description
<b>pclk</b>	Input	APB interface clock
<b>presetn</b>	Input	APB interface reset
<b>awvnet_0</b>	Input	AW channel virtual network ID
<b>wvnet_0</b>	Input	W channel virtual network ID
<b>arvnet_0</b>	Input	AR channel virtual network ID
<b>cwakeup_s0</b>	Input	Wakeup signal for the <b>s0_clk</b> clock domain
<b>cwakeup_m0</b>	Input	Wakeup signal for the <b>q0_clk</b> and <b>m0_clk</b> clock domains
<b>awap_0</b>	Input	Write auto-precharge policy signal
<b>arap_0</b>	Input	Read auto-precharge policy signal
<b>user_status</b>	Input	User-defined inputs
<b>user_config0</b>	Output	User-defined outputs
<b>user_config1</b>	Output	User-defined outputs
<b>ecc_int</b>	Output	Combined ECC interrupt
<b>ecc_sec_int</b>	Output	Single error corrected interrupt
<b>ecc_ded_int</b>	Output	Double error detected interrupt
<b>ecc_overflow_int</b>	Output	Error detection overflow interrupt
<b>ev_bus_valid_s0</b>	Output	System interface PMU event valid bus
<b>ev_bus_payload_s0</b>	Output	System interface PMU event payload bus
<b>ev_bus_valid_m0</b>	Output	Memory interface PMU event valid bus
<b>ev_bus_payload_m0</b>	Output	Memory interface PMU event payload bus
<b>dftse</b>	Input	DFT scan enable
<b>rst_bypass</b>	Input	DFT reset bypass

# Appendix B

## Revisions

This appendix describes the technical changes between released issues of this book.

**Table B-1 Issue A**

<b>Change</b>	<b>Location</b>	<b>Affects</b>
First release	-	-