

Arm® Cortex®-A72 MPCore Processor Cryptography Extension

Revision: r1p0

Technical Reference Manual



Arm® Cortex®-A72 MPCore Processor Cryptography Extension

Technical Reference Manual

Copyright © 2014–2016, 2018 Arm Limited or its affiliates. All rights reserved.

Release Information

Document History

Issue	Date	Confidentiality	Change
0000-01	23 October 2014	Confidential	First release for r0p0.
0001-02	20 February 2015	Confidential	First release for r0p1.
0002-03	05 June 2015	Confidential	First release for r0p2.
0002-04	17 December 2015	Non-Confidential	Second release for r0p2.
0003-05	22 April 2016	Non-Confidential	First release for r0p3.
0100-00	23 July 2018	Non-Confidential	First release for r1p0.

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 click through or signed written agreement covering this document with Arm, then the click through or 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.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US 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/company/policies/trademarks>.

Copyright © 2014–2016, 2018 Arm Limited (or its affiliates). All rights reserved.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

LES-PRE-20349

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

**Arm® Cortex®-A72 MPCore Processor Cryptography
Extension Technical Reference Manual**

	Preface	
	<i>About this book</i>	6
	<i>Feedback</i>	8
Chapter 1	Introduction	
	1.1 <i>About the Cortex-A72 processor Cryptography engine</i>	1-10
	1.2 <i>Product revisions</i>	1-11
Chapter 2	Programmers Model	
	2.1 <i>About the programmers model</i>	2-13
Appendix A	Revisions	
	A.1 <i>Revisions</i>	Appx-A-15

Preface

This preface introduces the *Arm® Cortex®-A72 MPCore Processor Cryptography Extension Technical Reference Manual*.

It contains the following:

- [About this book](#) on page 6.
- [Feedback](#) on page 8.

About this book

This document describes the instructions for the Cortex-A72 processor Cryptography extensions.

Product revision status

The *mpn* 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 system designers, system integrators, and programmers who are designing or programming a System-on-Chip (SoC) that uses the Cortex-A72 processor with the optional Cryptography Extension.

Using this book

This book is organized into the following chapters:

Chapter 1 Introduction

This chapter introduces the Cryptography Extensions instructions for the Cortex-A72 processor and its features.

Chapter 2 Programmers Model

This chapter describes the registers of the Cryptography engine and provides information for programming the engine.

Appendix A 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.

<and>

Encloses replaceable terms for assembler syntax where they appear in code or code fragments.
For example:

```
MRC p15, 0, <Rd>, <CRn>, <CRm>, <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.

Additional reading

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

Arm publications

- *Arm® Cortex®-A72 MPCore Processor Technical Reference Manual* (100095).
- *Arm® Architecture Reference Manual Arm®v8, for Arm®v8-A architecture profile* (DDI 0487).

The following confidential books are only available to licensees:

- *Arm® Cortex®-A72 MPCore Processor Configuration and Sign-off Guide* (100098).
- *Arm® Cortex®-A72 MPCore Processor Integration Manual* (100096).

Other publications

- *Advanced Encryption Standard* (FIPS 197, November 2001).
- *Secure Hash Standard (SHS)* (FIPS 180-4, March 2012).

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 Cortex-A72 MPCore Processor Cryptography Extension Technical Reference Manual*.
- The number 100097_0100_00_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 introduces the Cryptography Extensions instructions for the Cortex-A72 processor and its features.

It contains the following sections:

- [1.1 About the Cortex-A72 processor Cryptography engine](#) on page 1-10.
- [1.2 Product revisions](#) on page 1-11.

1.1 About the Cortex-A72 processor Cryptography engine

The Cortex-A72 processor Cryptography engine supports the Armv8 Cryptographic Extension. The Cryptographic Extension adds new instructions that the Advanced SIMD can use to accelerate the execution of AES, SHA1, and SHA2-256 algorithms.

The following table lists the instructions for AES. See the *Arm® Architecture Reference Manual, Arm®v8, for Arm®v8-A architecture profile* for more information.

Table 1-1 AES instructions

Mnemonic	Instruction
AESD	AES single round decryption
AESE	AES single round encryption
AESIMC	AES inverse mix columns
AESMC	AES mix columns
VMULL ^a	Polynomial multiply long

The following table lists the instructions for SHA1 or SHA2-256. See the *Arm® Architecture Reference Manual, Arm®v8, for Arm®v8-A architecture profile* for more information.

Table 1-2 SHA1 and SHA2-256 instructions

Mnemonic	Instruction
SHA1C	SHA1 hash update accelerator, choose
SHA1H	SHA1 fixed rotate
SHA1M	SHA1 hash update accelerator, majority
SHA1P	SHA1 hash update accelerator, parity
SHA1SU0	SHA1 schedule update accelerator, first part
SHA1SU1	SHA1 schedule update accelerator, second part
SHA256H	SHA256 hash update accelerator
SHA256H2	SHA256 hash update accelerator, upper part
SHA256SU0	SHA256 schedule update accelerator, first part
SHA256SU1	SHA256 schedule update accelerator, second part

^a Polynomial 64-bit instruction.

1.2 Product revisions

This section describes the differences in functionality between product revisions.

- r0p0** First release.
- r0p1** No technical changes for cryptography extension.
- r0p2** No technical changes for cryptography extension.
- r0p3** No technical changes for cryptography extension.
- r1p0** No technical changes for cryptography extension.

Chapter 2

Programmers Model

This chapter describes the registers of the Cryptography engine and provides information for programming the engine.

It contains the following section:

- [2.1 About the programmers model on page 2-13.](#)

2.1 About the programmers model

The Cortex-A72 processor Cryptography engine implements the Cryptography Extensions described in the Armv8 architecture.

This section contains the following subsections:

- [2.1.1 Identifying the cryptography instructions implemented on page 2-13.](#)
- [2.1.2 Disabling the Cryptography engine on page 2-13.](#)

2.1.1 Identifying the cryptography instructions implemented

Software can read a register to identify the cryptography instructions that are implemented.

The register to read depends on the Execution state, as follows:

AArch32

To access the ID_ISAR5 in AArch32 state, read the register with:

```
MRC p15, 0, <Rt>, c0, c2, 5 ; Read AArch32 Instruction Set Attribute Register 5
```

AArch64

To access the ID_ISAR5_EL1 in AArch64 state, read the register with:

```
MRS <Rd>, ID_ISAR5_EL1 ; Read AArch32 Instruction Set Attribute Register 5
```

To access the ID_AA64ISAR0_EL1 in AArch64 state, read the register with:

```
MRS <Xt>, ID_AA64ISAR0_EL1 ; Read AArch64 Instruction Set Attribute Register 0
```

The following table lists the instruction identification registers for the Cryptography engine. See the *Arm® Cortex®-A72 MPCore Processor Technical Reference Manual* for more information about the registers.

Table 2-1 Cryptography engine register summary

Name	Execution state	Description
ID_ISAR5	AArch32	AArch32 Instruction Set Attribute Register 5
ID_ISAR5_EL1	AArch64	
ID_AA64ISAR0_EL1		AArch64 Instruction Set Attribute Register 0

2.1.2 Disabling the Cryptography engine

The **CRYPTODISABLE[N:0]** input controls whether the Cryptography engine is disabled for processor *N*. The processor only samples this signal during reset.

When **CRYPTODISABLE** is HIGH, executing a cryptography instruction results in an Undefined Instruction exception.

Appendix A

Revisions

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

It contains the following section:

- [A.1 Revisions on page Appx-A-15.](#)

A.1 Revisions

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

Table A-1 Issue 0000-01

Change	Location	Affects
First release	-	-

Table A-2 Differences between issue 0000-01 and issue 0001-02

Change	Location	Affects
No technical changes.	-	-

Table A-3 Differences between issue 0001-02 and issue 0002-03

Change	Location	Affects
No technical changes.	-	-

Table A-4 Differences between issue 0002-03 and issue 0002-04

Change	Location	Affects
No technical changes.	-	-

Table A-5 Differences between issue 0002-04 and issue 0003-05

Change	Location	Affects
No technical changes.	-	-

Table A-6 Differences between issue 0003-05 and issue 0100-00

Change	Location	Affects
Updated company name to Arm.	-	-
No technical changes.	-	-