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  Programmers Model

Appendix A  Revisions
Preface

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

It contains the following:

- *Feedback* on page 8.
About this book

This document describes the instructions for the processor Cryptography extensions.

Product revision status

The rmpn 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 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.
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*. 
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 number ARM 100097_0003_05_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 Cryptography Extensions. The Cryptography Extensions add 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, ARMv8, for ARMv8-A architecture profile* for more information.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>AESD</td>
<td>AES single round decryption</td>
</tr>
<tr>
<td>AESE</td>
<td>AES single round encryption</td>
</tr>
<tr>
<td>AESIMC</td>
<td>AES inverse mix columns</td>
</tr>
<tr>
<td>AESMC</td>
<td>AES mix columns</td>
</tr>
<tr>
<td>VMULLa</td>
<td>Polynomial multiply long</td>
</tr>
</tbody>
</table>

The following table lists the instructions for SHA1 or SHA2-256. See the *ARM Architecture Reference Manual, ARMv8, for ARMv8-A architecture profile* for more information.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHA1C</td>
<td>SHA1 hash update accelerator, choose</td>
</tr>
<tr>
<td>SHA1H</td>
<td>SHA1 fixed rotate</td>
</tr>
<tr>
<td>SHA1M</td>
<td>SHA1 hash update accelerator, majority</td>
</tr>
<tr>
<td>SHA1P</td>
<td>SHA1 hash update accelerator, parity</td>
</tr>
<tr>
<td>SHA1SU0</td>
<td>SHA1 schedule update accelerator, first part</td>
</tr>
<tr>
<td>SHA1SU1</td>
<td>SHA1 schedule update accelerator, second part</td>
</tr>
<tr>
<td>SHA256H</td>
<td>SHA256 hash update accelerator</td>
</tr>
<tr>
<td>SHA256H2</td>
<td>SHA256 hash update accelerator, upper part</td>
</tr>
<tr>
<td>SHA256SU0</td>
<td>SHA256 schedule update accelerator, first part</td>
</tr>
<tr>
<td>SHA256SU1</td>
<td>SHA256 schedule update accelerator, second part</td>
</tr>
</tbody>
</table>

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

It contains the following sections:

- 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:

\[
\text{MRC p15, 0, } <Rt>, c0, c2, 5 \; \text{; Read AArch32 Instruction Set Attribute Register 5}
\]

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

\[
\text{MRS } <Rd>, \text{ ID_ISAR5_EL1 ; Read AArch32 Instruction Set Attribute Register 5}
\]

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

\[
\text{MRS } <Xt>, \text{ ID_AA64ISAR0_EL1 ; Read AArch64 Instruction Set Attribute Register 0}
\]

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>
<thead>
<tr>
<th>Name</th>
<th>Execution state</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID_ISAR5</td>
<td>AArch32</td>
<td>AArch32 Instruction Set Attribute Register 5</td>
</tr>
<tr>
<td>ID_ISAR5_EL1</td>
<td>AArch64</td>
<td>AArch64 Instruction Set Attribute Register 5</td>
</tr>
<tr>
<td>ID_AA64ISAR0_EL1</td>
<td>AArch64</td>
<td>AArch64 Instruction Set Attribute Register 0</td>
</tr>
</tbody>
</table>

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 sections:
### A.1 Revisions

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

#### Table A-1 Issue 01

<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 A-2 Differences between issue 01 and issue 02

<table>
<thead>
<tr>
<th>Change</th>
<th>Location</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>No technical changes.</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

#### Table A-3 Differences between issue 02 and issue 03

<table>
<thead>
<tr>
<th>Change</th>
<th>Location</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>No technical changes.</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Change</th>
<th>Location</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>No technical changes.</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

#### Table A-5 Differences between issue 04 and issue 05

<table>
<thead>
<tr>
<th>Change</th>
<th>Location</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>No technical changes.</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>