## Contents

**Arm® Cortex®-A76 Core Cryptographic Extension Technical Reference Manual**

### Preface

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About this book</td>
<td>6</td>
</tr>
<tr>
<td>Feedback</td>
<td>8</td>
</tr>
</tbody>
</table>

### Chapter 1 Functional description

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 About the Cryptographic Extension</td>
<td>1-10</td>
</tr>
<tr>
<td>1.2 Revisions</td>
<td>1-11</td>
</tr>
</tbody>
</table>

### Chapter 2 Register descriptions

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Identifying the cryptographic instructions implemented</td>
<td>2-13</td>
</tr>
<tr>
<td>2.2 Disabling the Cryptographic Extension</td>
<td>2-14</td>
</tr>
<tr>
<td>2.3 Register summary</td>
<td>2-15</td>
</tr>
<tr>
<td>2.4 ID_AA64ISAR0_EL1, AArch64 Instruction Set Attribute Register 0, EL1</td>
<td>2-16</td>
</tr>
<tr>
<td>2.5 ID_ISAR5_EL1, AArch32 Instruction Set Attribute Register 5, EL1</td>
<td>2-18</td>
</tr>
</tbody>
</table>

### Appendix A Revisions

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1 Revisions</td>
<td>Appx-A-21</td>
</tr>
</tbody>
</table>
Preface

This preface introduces the Arm® Cortex®-A76 Core Cryptographic 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 optional cryptographic features of the core. It includes descriptions of the registers used by the Cryptographic Extension.

Product revision status

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

- \texttt{rm} Identifies the major revision of the product, for example, \texttt{r1}.
- \texttt{pn} Identifies the minor revision or modification status of the product, for example, \texttt{p2}.

Intended audience

This manual is for system designers, system integrators, and programmers who are designing or programming a System-on-Chip (SoC) that uses the Cortex\textsuperscript{®}-A76 core with the optional Cryptographic Extension.

Using this book

This book is organized into the following chapters:

- **Chapter 1 Functional description**
  This chapter describes the Cortex-A76 core Cryptographic Extension.

- **Chapter 2 Register descriptions**
  This chapter describes the Cryptographic Extension registers.

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

Glossary

The Arm\textsuperscript{®} 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\textsuperscript{®} 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.

**Additional reading**

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

**Arm publications**

- *Arm® Cortex®-A76 Core Integration Manual* (100800).

**Other publications**

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 100801_0400_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
Functional description

This chapter describes the Cortex-A76 core Cryptographic Extension.

It contains the following sections:

- 1.1 About the Cryptographic Extension on page 1-10.
- 1.2 Revisions on page 1-11.
1.1 About the Cryptographic Extension

The Cortex-A76 core Cryptographic Extension supports the Armv8-A Cryptographic Extension.

The Cryptographic Extension adds new A64, A32, and T32 instructions to Advanced SIMD that accelerate Advanced Encryption Standard (AES) encryption and decryption. It also adds instructions to implement the Secure Hash Algorithm (SHA) functions SHA-1, SHA-224, and SHA-256.

Note

The optional Cryptographic Extension is not included in the base product. Arm supplies the Cryptographic Extension only under an additional license to the Cortex-A76 core.
1.2 Revisions

This section describes the differences in functionality between product revisions.

- r0p0: First release.
- r1p0: No functional changes.
- r2p0: No functional changes.
- r3p0: No functional changes.
- r3p1: No functional changes.
- r4p0: No functional changes.
Chapter 2
Register descriptions

This chapter describes the Cryptographic Extension registers.

It contains the following sections:

• 2.1 Identifying the cryptographic instructions implemented on page 2-13.
• 2.2 Disabling the Cryptographic Extension on page 2-14.
• 2.3 Register summary on page 2-15.
• 2.4 ID_AA64ISAR0_EL1, AArch64 Instruction Set Attribute Register 0, EL1 on page 2-16.
• 2.5 ID_ISAR5_EL1, AArch32 Instruction Set Attribute Register 5, EL1 on page 2-18.
2.1 Identifying the cryptographic instructions implemented

Software can identify the cryptographic instructions that are implemented by reading two registers.

The two registers are:

- ID_AA64ISAR0_EL1 in the AArch64 execution state.
- ID_ISAR5_EL1 in the AArch64 execution state.
2.2 Disabling the Cryptographic Extension

To disable the Cryptographic Extension, assert the CRYPTODISABLE input signal that applies to all the Cortex-A76 cores present in a cluster. This signal is sampled only during reset of the cores.

When CRYPTODISABLE is asserted:
• Executing a cryptographic instruction results in an UNDEFINED exception.
• The ID registers described in Table 2-1 Cryptographic Extension register summary on page 2-15 indicate that the Cryptographic Extension is not implemented.
2.3 Register summary

The core has two instruction identification registers. Each register has a specific purpose, usage constraints, configurations, and attributes.

The following table lists the instruction identification registers for the Cortex-A76 core Cryptographic Extension.

<table>
<thead>
<tr>
<th>Name</th>
<th>Execution state</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID_AA64ISAR0_EL1</td>
<td>AArch64</td>
<td>See 2.4 ID_AA64ISAR0_EL1, AArch64 Instruction Set Attribute Register 0, EL1 on page 2-16.</td>
</tr>
<tr>
<td>ID_ISAR5_EL1</td>
<td>AArch64</td>
<td>See 2.5 ID_ISAR5_EL1, AArch32 Instruction Set Attribute Register 5, EL1 on page 2-18.</td>
</tr>
</tbody>
</table>
2.4 ID_AA64ISAR0_EL1, AArch64 Instruction Set Attribute Register 0, EL1

The ID_AA64ISAR0_EL1 provides information about the instructions implemented in AArch64 state, including the instructions provided by the Cryptographic Extension.

**Bit field descriptions**

ID_AA64ISAR0_EL1 is a 64-bit register.

<table>
<thead>
<tr>
<th>Bit Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>63-48</td>
<td>RES0</td>
</tr>
<tr>
<td>47-44</td>
<td>DP</td>
</tr>
<tr>
<td>43-32</td>
<td>RDM</td>
</tr>
<tr>
<td>31-24</td>
<td>Atomic</td>
</tr>
<tr>
<td>23-19</td>
<td>CRC32</td>
</tr>
<tr>
<td>18-12</td>
<td>SHA2</td>
</tr>
<tr>
<td>11-8</td>
<td>SHA1</td>
</tr>
<tr>
<td>7-4</td>
<td>AES</td>
</tr>
</tbody>
</table>

**Figure 2-1 ID_AA64ISAR0_EL1 bit assignments**

- **RES0**, [63:48]
  - RES0 Reserved.

- **DP**, [47:44]
  - Indicates whether Dot Product support instructions are implemented.
  - 0x1 UDOT, SDOT instructions are implemented.

- **RES0**, [43:32]
  - RES0 Reserved.

- **RDM**, [31:28]
  - Indicates whether Rounding Double Multiply (RDM) instructions are implemented. The value is:
  - 0x1 SQRDMLAH and SQRDMLSH instructions are implemented.

- **[27:24]**
  - RES0 Reserved.

- **Atomic**, [23:20]
  - Indicates whether atomic instructions are implemented. The value is:
  - 0x2 LDADD, LDCLR, LDEOR, LDSSET, LDSMAX, LDSMIN, LDUMAX, LDUMIN, CAS, CASP, and SWP instructions are implemented.

- **CRC32**, [19:16]
  - Indicates whether CRC32 instructions are implemented. The value is:
  - 0x1 CRC32 instructions are implemented.

- **SHA2**, [15:12]
  - Indicates whether SHA2 instructions are implemented. The possible values are:
  - 0x0 No SHA2 instructions are implemented. This is the value if the core implementation does not include the Cryptographic Extension.
  - 0x1 SHA256H, SHA256H2, SHA256U0, and SHA256U1 are implemented. This is the value if the core implementation includes the Cryptographic Extension.

- **SHA1**, [11:8]
Indicates whether SHA1 instructions are implemented. The possible values are:

0x0  No SHA1 instructions are implemented. This is the value if the core implementation does not include the Cryptographic Extension.

0x1  SHA1C, SHA1P, SHA1M, SHA1SU0, and SHA1SU1 are implemented. This is the value if the core implementation includes the Cryptographic Extension.

AES, [7:4]
Indicates whether AES instructions are implemented. The possible values are:

0x0  No AES instructions implemented. This is the value if the core implementation does not include the Cryptographic Extension.

0x2  AESE, AESD, AESMC, and AESIMC are implemented, plus PMULL and PMUL2 instructions operating on 64-bit data. This is the value if the core implementation includes the Cryptographic Extension.

[3:0]
RES0  Reserved.

Configurations
ID_AA64ISAR0_EL1 is architecturally mapped to external register ID_AA64ISAR0.

Usage constraints
Accessing the ID_AA64ISAR0_EL1
To access the ID_AA64ISAR0_EL1:

MRS <Xt>, ID_AA64ISAR0_EL1 ; Read ID_AA64ISAR0_EL1 into Xt

Register access is encoded as follows:

Table 2-2  ID_AA64ISAR0_EL1 access encoding

<table>
<thead>
<tr>
<th>op0</th>
<th>op1</th>
<th>CRn</th>
<th>CRm</th>
<th>op2</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>000</td>
<td>0000</td>
<td>0110</td>
<td>000</td>
</tr>
</tbody>
</table>

Accessibility
This register is accessible as follows:

<table>
<thead>
<tr>
<th>EL0</th>
<th>EL3</th>
<th>EL1</th>
<th>EL2</th>
<th>EL3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(NS)</td>
<td>(S)</td>
<td>(SCR.NS = 1)</td>
<td>(SCR.NS = 0)</td>
<td></td>
</tr>
<tr>
<td>RO</td>
<td>RO</td>
<td>RO</td>
<td>RO</td>
<td>RO</td>
</tr>
</tbody>
</table>
2.5 ID_ISAR5_EL1, AArch32 Instruction Set Attribute Register 5, EL1

The AArch64 register ID_ISAR5_EL1 provides information about the instructions implemented in AArch32 state, including the instructions provided by the optional Cryptographic Extension.

**Bit field descriptions**

ID_ISAR5_EL1 is a 32-bit register.

<table>
<thead>
<tr>
<th>Bit Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[31:29]</td>
<td>RDM</td>
</tr>
<tr>
<td>[28:26]</td>
<td>CRC32</td>
</tr>
<tr>
<td>[25:23]</td>
<td>SHA2</td>
</tr>
<tr>
<td>[22:20]</td>
<td>SHA1</td>
</tr>
<tr>
<td>[19:17]</td>
<td>AES</td>
</tr>
<tr>
<td>[16:14]</td>
<td>SEVL</td>
</tr>
</tbody>
</table>

![Figure 2-2 ID_ISAR5_EL1 bit assignments](image)

- **RDM, [27:24]**
  - Indicates whether RDM instructions are implemented. The value is:
    - 0x1: SQRDMLH and SQRDMLSH instructions are implemented.

- **CRC32, [19:16]**
  - Indicates whether CRC32 instructions are implemented in AArch32 state. The value is:
    - 0x1: CRC32 instructions are implemented.

- **SHA2, [15:12]**
  - Indicates whether SHA2 instructions are implemented in AArch32 state. The possible values are:
    - 0x0: Cryptographic Extension is not implemented or is disabled.
    - 0x1: SHA256H, SHA256H2, SHA256SU0, and SHA256SU1 instructions are implemented.

- **SHA1, [11:8]**
  - Indicates whether SHA1 instructions are implemented in AArch32 state. The possible values are:
    - 0x0: Cryptographic Extension is not implemented or is disabled.
    - 0x1: SHA1C, SHA1P, SHA1M, SHA1H, SHA1SU0, and SHA1SU1 instructions are implemented.

- **AES, [7:4]**
  - Indicates whether AES instructions are implemented in AArch32 state. The possible values are:
    - 0x0: Cryptographic Extension is not implemented or is disabled.
    - 0x2: AESE, AESD, AESMC, and AESIMC are implemented, plus PMULL and PMULL2 instructions operating on 64-bit data.
SEVL, [3:0]
Indicates whether the SEVL instruction is implemented. The value is:
\[0x1 \quad \text{SEVL implemented to send event local.}\]

Configurations
This register has no configuration options.

Usage constraints

Accessing the ID_ISAR5_EL1
To access the ID_ISAR5_EL1:
\[
\text{MRS } \langle X_t \rangle, \text{ ID_ISAR5_EL1 } \ ; \ \text{Read ID_ISAR5_EL1 into } X_t
\]

Register access is encoded as follows:

Table 2-3  ID_ISAR5_EL1 access encoding

<table>
<thead>
<tr>
<th>op0</th>
<th>op1</th>
<th>CRn</th>
<th>CRm</th>
<th>op2</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>000</td>
<td>0000</td>
<td>0010</td>
<td>101</td>
</tr>
</tbody>
</table>

Accessibility
This register is accessible as follows:

<table>
<thead>
<tr>
<th>EL0</th>
<th>EL1 (NS)</th>
<th>EL1 (S)</th>
<th>EL2 (SCR.NS = 1)</th>
<th>EL3 (SCR.NS = 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>RO</td>
<td>RO</td>
<td>RO</td>
<td>RO</td>
</tr>
</tbody>
</table>

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-21.
A.1 Revisions

This section describes the technical changes between released issues of this document.

Table A-1 Issue 0000-00

<table>
<thead>
<tr>
<th>Change</th>
<th>Location</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>First draft for r0p0</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table A-2 Differences between issue 0000-00 and issue 0000-01

<table>
<thead>
<tr>
<th>Change</th>
<th>Location</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>First release for r0p0. No technical changes.</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table A-3 Differences between issue 0000-01 and issue 0100-00

<table>
<thead>
<tr>
<th>Change</th>
<th>Location</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>First release for r1p0. No technical changes.</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table A-4 Differences between issue 0100-00 and issue 0200-00

<table>
<thead>
<tr>
<th>Change</th>
<th>Location</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>First release for r2p0. No technical changes.</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table A-5 Differences between issue 0200-00 and issue 0300-00

<table>
<thead>
<tr>
<th>Change</th>
<th>Location</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>First release for r3p0. No technical changes.</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table A-6 Differences between issue 0300-00 and issue 0301-00

<table>
<thead>
<tr>
<th>Change</th>
<th>Location</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>First release for r3p1. No technical changes.</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table A-7 Differences between issue 0301-00 and issue 0400-00

<table>
<thead>
<tr>
<th>Change</th>
<th>Location</th>
<th>Affects</th>
</tr>
</thead>
<tbody>
<tr>
<td>First release for r4p0. No technical changes.</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>