



# Arm<sup>®</sup> Cortex-X4 Core Cryptographic Extension

Revision: r0p2

## Technical Reference Manual

**Non-Confidential**

**Issue 05**

Copyright © 2021–2023 Arm Limited (or its affiliates). 102486\_0002\_05\_en  
All rights reserved.



# Arm® Cortex-X4 Core Cryptographic Extension

## Technical Reference Manual

Copyright © 2021–2023 Arm Limited (or its affiliates). All rights reserved.

## Release Information

### Document history

Issue	Date	Confidentiality	Change
0000-01	18 November 2021	Confidential	First beta release for r0p0
0000-02	8 April 2022	Confidential	First limited access release for r0p0
0001-03	28 July 2022	Confidential	First early access release for r0p1
0001-04	29 May 2023	Non-Confidential	Second early access release for r0p1
0002-05	30 November 2023	Non-Confidential	First release for r0p2

## 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, 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.

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 affiliates) 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 <https://www.arm.com/company/policies/trademarks>.

Copyright © 2021–2023 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|version 21.0)

## 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.

## Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on <https://support.developer.arm.com>.

To provide feedback on the document, fill the following survey: <https://developer.arm.com/documentation-feedback-survey>.

## Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

This document includes language that can be offensive. We will replace this language in a future issue of this document.

To report offensive language in this document, email [terms@arm.com](mailto:terms@arm.com).

# Contents

- 1. Introduction..... 6**
  - 1.1 Product revision status..... 6
  - 1.2 Intended audience..... 6
  - 1.3 Conventions..... 6
  - 1.4 Useful resources..... 8
- 2. Cryptographic Extension support in the Cortex-X4 core..... 10**
  - 2.1 Disabling the Cryptographic Extension..... 10
  - 2.2 Product Revisions..... 11
- 3. AArch64 instruction identification system registers..... 12**
  - 3.1 Cryptographic Extensions register summary..... 12
  - 3.2 ID\_AA64ISAR0\_EL1, AArch64 Instruction Set Attribute Register 0..... 12
  - 3.3 ID\_AA64ZFR0\_EL1, SVE Feature ID register 0..... 16
- A. Document revisions..... 20**
  - A.1 Revisions..... 20

# 1. Introduction

## 1.1 Product revision status

The  $r_xp_y$  identifier indicates the revision status of the product described in this manual, for example,  $r1p2$ , where:

<b><math>r_x</math></b>	Identifies the major revision of the product, for example, $r1$ .
<b><math>p_y</math></b>	Identifies the minor revision or modification status of the product, for example, $p2$ .

## 1.2 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-X4 core with the optional Cryptographic Extension.

## 1.3 Conventions

The following subsections describe conventions used in Arm documents.

### 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: [developer.arm.com/glossary](https://developer.arm.com/glossary).

Convention	Use
<i>italic</i>	Citations.
<b>bold</b>	Terms in descriptive lists, where appropriate.
monospace	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
monospace <u>underline</u>	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.

Convention	Use
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments.  For example:  <pre>MRC p15, 0, &lt;Rd&gt;, &lt;CRn&gt;, &lt;CRm&gt;, &lt;Opcode_2&gt;</pre>
<b>SMALL CAPITALS</b>	Terms that have specific technical meanings as defined in the <i>Arm® Glossary</i> . For example, <b>IMPLEMENTATION DEFINED</b> , <b>IMPLEMENTATION SPECIFIC</b> , <b>UNKNOWN</b> , and <b>UNPREDICTABLE</b> .



Recommendations. Not following these recommendations might lead to system failure or damage.



Requirements for the system. Not following these requirements might result in system failure or damage.



Requirements for the system. Not following these requirements will result in system failure or damage.



An important piece of information that needs your attention.



A useful tip that might make it easier, better or faster to perform a task.



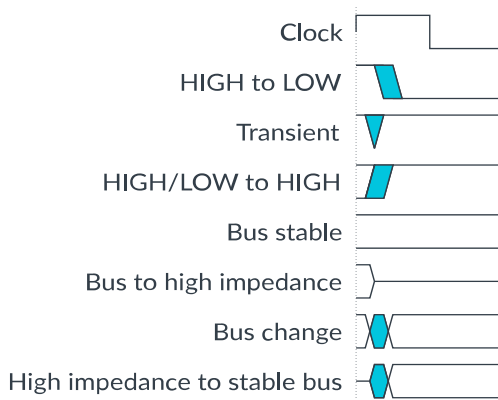
A reminder of something important that relates to the information you are reading.

## Timing diagrams

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

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

**Figure 1-1: Key to timing diagram conventions**



## Signals

The signal conventions are:

### Signal level

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

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

### Lowercase n

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

## 1.4 Useful resources

This document contains information that is specific to this product. See the following resources for other useful information.

Access to Arm documents depends on their confidentiality:

- Non-Confidential documents are available at [developer.arm.com/documentation](https://developer.arm.com/documentation). Each document link in the following tables goes to the online version of the document.
- Confidential documents are available to licensees only through the product package.

Arm product resources	Document ID	Confidentiality
Arm® Cortex-X4 Core Configuration and Integration Manual	102485	Confidential
Arm® Cortex-X4 Core Technical Reference Manual	102484	Non-Confidential



Arm architecture and specifications	Document ID	Confidentiality
<i>Arm® Architecture Reference Manual for A-profile architecture</i>	DDI 0487	Non-Confidential
<i>Arm® Architecture Reference Manual Supplement Armv9, for Armv9-A architecture profile</i>	DDI 0608	Non-Confidential



Arm tests its PDFs only in Adobe Acrobat and Acrobat Reader. Arm cannot guarantee the quality of its documents when used with any other PDF reader.

Adobe PDF reader products can be downloaded at <http://www.adobe.com>.

---

## 2. Cryptographic Extension support in the Cortex-X4 core

The Cortex-X4 core supports the optional Arm® Cryptographic Extension.

The Arm® Cryptographic Extension adds A64 instructions to Advanced SIMD to:

- Accelerate *Advanced Encryption Standard* (AES) encryption and decryption
- Implement the *Secure Hash Algorithm* (SHA) functions
- Perform *Polynomial Multiply Long* (PMULL) instructions

### Supported features

The Arm® Cryptographic Extension supports the following features:

**Table 2-1: Features supported by the Arm® Cryptographic Extension**

Feature	Description	Architecture version
FEAT_AES	Advanced SIMD AES instructions	Arm®v8.0
FEAT_PMULL	Advanced SIMD PMULL instructions	
FEAT_SHA1	Advanced SIMD SHA1 instructions	
FEAT_SHA256	Advanced SIMD SHA256 instructions	
FEAT_SHA512	Advanced SIMD SHA512 instructions	Arm®v8.2
FEAT_SHA3	Advanced SIMD EOR3, RAX1, XAR, and BCAX instructions	
FEAT_SM3	Advanced SIMD SM3 instructions	
FEAT_SM4	Advanced SIMD SM4 instructions	
FEAT_SVE_AES	SVE AES instructions	Arm®v9.0
FEAT_SVE_PMULL128	SVE PMULL instructions	
FEAT_SVE_SHA3	SVE SHA3 instructions	
FEAT_SVE_SM4	SVE SM4 instructions	

### 2.1 Disabling the Cryptographic Extension

Disabling the Cryptographic Extension applies to all Cortex-X4 cores in a cluster.

To disable the Cryptographic Extension, assert the CRYPTODISABLE signal.

When the CRYPTODISABLE signal is asserted:

- Executing a cryptographic instruction results in an **UNDEFINED** exception.
- ID\_AA64ISAR0\_EL1 and ID\_AA64ZFR0\_EL1 indicate that the Cryptographic Extension is not implemented.

## Related information

[3.2 ID\\_AA64ISAR0\\_EL1, AArch64 Instruction Set Attribute Register 0](#) on page 12

[3.3 ID\\_AA64ZFR0\\_EL1, SVE Feature ID register 0](#) on page 16

## 2.2 Product Revisions

The following table indicates the main differences in functionality between product revisions.

**Table 2-2: Product revisions**

Revision	Notes
r0p0	First limited access release
r0p1	First early access release. Bug fixes only.
r0p2	First release. Bug fixes only.

Changes in functionality that have an impact on the documentation also appear in [A.1 Revisions](#) on page 20.

## 3. AArch64 instruction identification system registers

This chapter describes the ID\_AA64ISAR0\_EL1 and ID\_AA64ZFR0\_EL1 registers. These identification registers provide information about the instructions implemented in the Cortex-X4 core, including the instructions provided by the Cryptographic Extension.

### 3.1 Cryptographic Extensions register summary

Software can identify the cryptographic instructions that are implemented in the Cortex-X4 core by reading the identification registers.

The following table shows the identification registers for the Cortex-X4 core Cryptographic Extension.

**Table 3-1: Cryptographic Extension register summary**

Name	Description
ID_AA64ISAR0_EL1	See 3.2 ID_AA64ISAR0_EL1, AArch64 Instruction Set Attribute Register 0 on page 12
ID_AA64ZFR0_EL1	See 3.3 ID_AA64ZFR0_EL1, SVE Feature ID register 0 on page 16

### 3.2 ID\_AA64ISAR0\_EL1, AArch64 Instruction Set Attribute Register 0

Provides information about the instructions implemented in AArch64 state.

For general information about the interpretation of the ID registers, see *Principles of the ID scheme for fields in ID registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

#### Configurations

This register is available in all configurations.

#### Attributes

##### Width

64

##### Functional group

Identification registers

##### Access type

See bit descriptions

## Reset value

0000	0010	0010	0001	0001	xxxx	xxxx	xxxx	0001	0000	0010	0001	xxxx	xxxx	xxxx	xxxx
63	59	55	51	47	43	39	35	31	27	23	19	15	11	7	3
															0



Where the reset reads xxxx, see individual bits.

## Bit descriptions

Figure 3-1: AArch64\_id\_aa64isar0\_el1 bit assignments

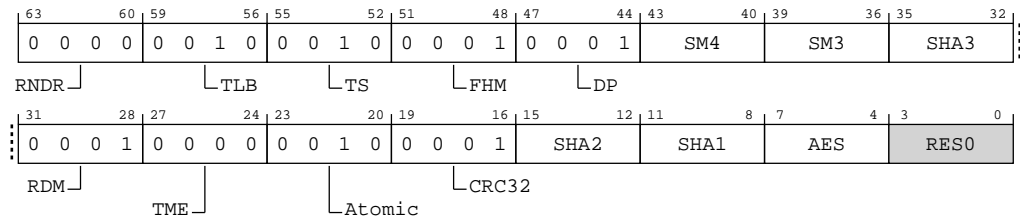


Table 3-2: ID\_AA64ISAR0\_EL1 bit descriptions

Bits	Name	Description	Reset
[63:60]	RNDR	Indicates support for Random Number instructions in AArch64 state.  When FEAT_RNG_TRAP is implemented, the value returned by a direct read of ID_AA64ISAR0_EL1.RNDR is further controlled by the value of AArch64-SCR_EL3.TRNDR.  Defined values are: <b>0b0000</b> No Random Number instructions are implemented.	0b0000
[59:56]	TLB	Indicates support for Outer Shareable and TLB range maintenance instructions. Defined values are: <b>0b0010</b> Outer Shareable and TLB range maintenance instructions are implemented.	0b0010
[55:52]	TS	Indicates support for flag manipulation instructions. Defined values are: <b>0b0010</b> CFINV, RMIF, SETF16, SETF8, AXFLAG, and XAFLAG instructions are implemented.	0b0010
[51:48]	FHM	Indicates support for FMLAL and FMLSL instructions. Defined values are: <b>0b0001</b> FMLAL and FMLSL instructions are implemented.	0b0001

Bits	Name	Description	Reset
[47:44]	DP	Indicates support for Dot Product instructions in AArch64 state. Defined values are:  <b>0b0001</b> UDOT and SDOT instructions implemented.	0b0001
[43:40]	SM4	Indicates support for SM4 instructions in AArch64 state. Defined values are:  <b>0b0000</b> When the Cryptographic Extension is not implemented or is disabled or the SM3/SM4 Cryptographic instructions are disabled, then SM4 instructions are not implemented.  <b>0b0001</b> When the Cryptographic Extension is implemented and the SM3/SM4 Cryptographic instructions are enabled, then SM4 instructions SM4E and SM4EKEY are implemented.	The reset values can be the following: 0b0000, 0b0001, respective to the value.
[39:36]	SM3	Indicates support for SM3 instructions in AArch64 state. Defined values are:  <b>0b0000</b> When the Cryptographic Extension is not implemented or is disabled or the SM3/SM4 Cryptographic instructions are disabled, then SM3 instructions are not implemented.  <b>0b0001</b> When the Cryptographic Extension is implemented and the SM3/SM4 Cryptographic instructions are enabled, then SM3 instructions SM3SS1, SM3TT1A, SM3TT1B, SM3TT2A, SM3TT2B, SM3PARTW1, and SM3PARTW2 are implemented.	The reset values can be the following: 0b0000, 0b0001, respective to the value.
[35:32]	SHA3	Indicates support for SHA3 instructions in AArch64 state. Defined values are:  <b>0b0000</b> When Cryptographic extensions are not implemented or disabled then SHA3 instructions are not implemented.  <b>0b0001</b> When Cryptographic extensions are implemented and enabled then SHA3 instructions EOR3, RAX1, XAR, and BCAX are implemented.	The reset values can be the following: 0b0000, 0b0001, respective to the value.
[31:28]	RDM	Indicates support for SQRDMLAH and SQRDMLSH instructions in AArch64 state. Defined values are:  <b>0b0001</b> SQRDMLAH and SQRDMLSH instructions implemented.	0b0001
[27:24]	TME	Indicates support for TME instructions. Defined values are:  <b>0b0000</b> TME instructions are not implemented.  <b>When PSTATE.EL IN {EL2, EL1}</b> Access to this field is: <b>RAZ/WI</b>  <b>When PSTATE.EL == EL1 &amp;&amp; EL2Enabled()</b> Access to this field is: <b>RAZ/WI</b>  <b>Otherwise</b> Access to this field is: <b>RO</b>	0b0000

Bits	Name	Description	Reset
[23:20]	Atomic	Indicates support for Atomic instructions in AArch64 state. Defined values are:  <b>0b0010</b> LDADD, LDCLR, LDEOR, LDSET, LDSMAX, LDSMIN, LDUMAX, LDUMIN, CAS, CASP, and SWP instructions implemented.	0b0010
[19:16]	CRC32	Indicates support for CRC32 instructions in AArch64 state. Defined values are:  <b>0b0001</b> CRC32B, CRC32H, CRC32W, CRC32X, CRC32CB, CRC32CH, CRC32CW, and CRC32CX instructions are implemented.	0b0001
[15:12]	SHA2	Indicates support for SHA2 instructions in AArch64 state. Defined values are:  <b>0b0000</b> When Cryptographic extensions are not implemented or disabled then SHA2 instructions are not implemented.  <b>0b0010</b> When Cryptographic extensions are implemented and enabled then SHA256H, SHA256H2, SHA256SU0, SHA256SU1, SHA512H, SHA512H2, SHA512SU0, and SHA512SU1 instructions are implemented.  When the CRYPTO configuration parameter is true and the CRYPTODISABLE input is low at reset Cryptographic Extensions are implemented	The reset values can be the following: 0b0000, 0b0010, respective to the value.
[11:8]	SHA1	Indicates support for SHA1 instructions in AArch64 state. Defined values are:  <b>0b0000</b> When Cryptographic extensions are not implemented or disabled then SHA1 instructions are not implemented.  <b>0b0001</b> When Cryptographic extensions are implemented and enabled then SHA1C, SHA1P, SHA1M, SHA1H, SHA1SU0, and SHA1SU1 instructions are implemented.  When the CRYPTO configuration parameter is true and the CRYPTODISABLE input is low at reset Cryptographic Extensions are implemented	The reset values can be the following: 0b0000, 0b0001, respective to the value.
[7:4]	AES	Indicates support for AES instructions in AArch64 state. Defined values are:  <b>0b0000</b> SVE2-AES instructions are not implemented. This value is reported when Cryptographic extensions are not implemented or are disabled.  <b>0b0010</b> SVE2 AESE, AESD, AESMC, and AESIMC instructions are implemented plus SVE2 PMULLB and PMULLT instructions with 64-bit source. This value is reported when Cryptographic extensions are implemented and enabled.  When the CRYPTO configuration parameter is true and the CRYPTODISABLE input is low at reset Cryptographic Extensions are implemented	The reset values can be the following: 0b0000, 0b0010, respective to the value.
[3:0]	RES0	Reserved	RES0

## Access

MRS <Xt>, ID\_AA64ISAR0\_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b0000	0b0110	0b000

## Accessibility

MRS <Xt>, ID\_AA64ISAR0\_EL1

```
if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TID3 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ID_AA64ISAR0_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = ID_AA64ISAR0_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = ID_AA64ISAR0_EL1;
```

## 3.3 ID\_AA64ZFR0\_EL1, SVE Feature ID register 0

Provides additional information about the implemented features of the AArch64 Scalable Vector Extension instruction set, when one or more of FEAT\_SVE and FEAT\_SME is implemented.

For general information about the interpretation of the ID registers, see *Principles of the ID scheme for fields in ID registers* in the [Arm® Architecture Reference Manual for A-profile architecture](#).

### Configurations



Prior to the introduction of the features described by this register, this register was unnamed and reserved, RES0 from EL1, EL2, and EL3.

If FEAT\_SME is implemented and FEAT\_SVE is not implemented, then SVE instructions can only be executed when the PE is in Streaming SVE mode and the instructions are legal to execute in Streaming SVE mode.

### Attributes

#### Width

64

#### Functional group

Identification registers

#### Access type

See bit descriptions

#### Reset value

xxxx	0000	0000	xxxx	0001	xxxx	xxxx	xxxx	xxxx	0000	0001	0001	xxxx	xxxx	xxxx	0001
63	59	55	51	47	43	39	35	31	27	23	19	15	11	7	3 0





Where the reset reads xxxx, see individual bits.

## Bit descriptions

Figure 3-2: AArch64\_id\_aa64zfr0\_el1 bit assignments

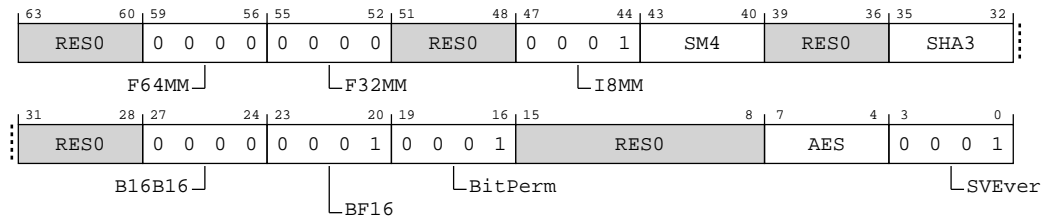


Table 3-4: ID\_AA64ZFR0\_EL1 bit descriptions

Bits	Name	Description	Reset
[63:60]	RES0	Reserved	RES0
[59:56]	F64MM	Indicates support for SVE FP64 double-precision floating-point matrix multiplication instructions. Defined values are:  <b>0b0000</b> Double-precision matrix multiplication and related SVE instructions are not implemented.	0b0000
[55:52]	F32MM	Indicates support for the SVE FP32 single-precision floating-point matrix multiplication instruction. Defined values are:  <b>0b0000</b> Single-precision matrix multiplication instruction is not implemented.	0b0000
[51:48]	RES0	Reserved	RES0
[47:44]	I8MM	Indicates support for SVE Int8 matrix multiplication instructions. Defined values are:  <b>0b0001</b> SVE SMMLA, SUDOT, UMMLA, USMMLA, and USDOT instructions are implemented.	0b0001
[43:40]	SM4	Indicates support for SVE SM4 instructions. Defined values are:  <b>0b0000</b> SVE2 SM4 instructions are not implemented. This value is reported when the Cryptographic Extension is not implemented or is disabled, or SM3/SM4 Cryptographic extensions are not implemented or are disabled.  <b>0b0001</b> SVE2 SM4E and SM4EKEY instructions are implemented. This value is reported when the Cryptographic Extension is implemented and SM3/SM4 Cryptographic instructions are enabled.	The reset values can be the following: 0b0000, 0b0001, respective to the value.
[39:36]	RES0	Reserved	RES0

Bits	Name	Description	Reset
[35:32]	SHA3	Indicates support for the SVE SHA3 instructions. Defined values are:  <b>0b0000</b> SVE2 SHA-3 instructions are not implemented. This value is reported when Cryptographic extensions are not implemented or are disabled.  <b>0b0001</b> SVE2 RAX1 instruction is implemented. This value is reported when Cryptographic extensions are implemented and enabled.	The reset values can be the following: 0b0000, 0b0001, respective to the value.
[31:28]	RES0	Reserved	RES0
[27:24]	B16B16	Indicates support for SVE2.1 non-widening BFloat16 instructions. Defined values are:  <b>0b0000</b> SVE2.1 non-widening BFloat16 instructions are not implemented.	0b0000
[23:20]	BF16	Indicates support for SVE BFloat16 instructions. Defined values are:  <b>0b0001</b> SVE BFCVT, BFCVTNT, BFDOT, BFMLALB, BFMLALT, and BFMMLA instructions are implemented.	0b0001
[19:16]	BitPerm	Indicates support for SVE bit permute instructions. Defined values are:  <b>0b0001</b> SVE BDEP, BEXT, and BGRP instructions are implemented.	0b0001
[15:8]	RES0	Reserved	RES0
[7:4]	AES	Indicates support for SVE AES instructions. Defined values are:  <b>0b0000</b> SVE2-AES instructions are not implemented. This value is reported when Cryptographic extensions are not implemented or are disabled.  <b>0b0010</b> SVE2 AESE, AESD, AESMC, and AESIMC instructions are implemented plus SVE2 PMULLB and PMULLT instructions with 64-bit source. This value is reported when Cryptographic extensions are implemented and enabled.	The reset values can be the following: 0b0000, 0b0010, respective to the value.
[3:0]	SVEver	Indicates support for SVE instructions when one or more of FEAT_SME and FEAT_SVE is implemented. Defined values are:  <b>0b0000</b> The SVE instructions are implemented.  <b>0b0001</b> As 0b0000, and adds the mandatory SVE2 instructions.  For this product, the selected value is 0b0001.	0b0001

## Access

MRS <Xt>, ID\_AA64ZFR0\_EL1

op0	op1	CRn	CRm	op2
0b11	0b000	0b0000	0b0100	0b100

## Accessibility

MRS <Xt>, ID\_AA64ZFR0\_EL1

```
if PSTATE.EL == EL0 then
    if EL2Enabled() && HCR_EL2.TGE == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        AArch64.SystemAccessTrap(EL1, 0x18);
elseif PSTATE.EL == EL1 then
    if EL2Enabled() && HCR_EL2.TID3 == '1' then
        AArch64.SystemAccessTrap(EL2, 0x18);
    else
        X[t, 64] = ID_AA64ZFR0_EL1;
elseif PSTATE.EL == EL2 then
    X[t, 64] = ID_AA64ZFR0_EL1;
elseif PSTATE.EL == EL3 then
    X[t, 64] = ID_AA64ZFR0_EL1;
```

# Appendix A Document revisions

This appendix records the changes between released issues of this document.

## A.1 Revisions

The first table is for the first release.

The first table is for the first release. Then, each table compares the new issue of the book with the last released issue of the book. Release numbers match the revision history in [Release Information](#) on page 2.

**Table A-1: Issue 0000-01**

Change	Location
First Beta release for r0p0	-

**Table A-2: Differences between issue 0000-01 and issue 0000-02**

Change	Location
First limited access release for r0p0	-
Editorial changes	Throughout document
Removed information about long polynomials from PMULL description	<a href="#">2. Cryptographic Extension support in the Cortex-X4 core</a> on page 10
Additional information about how to implement the Cryptographic Extension was added to SM3, SM4, and SHA3 bit descriptions	<a href="#">3.2 ID_AA64ISAR0_EL1, AArch64 Instruction Set Attribute Register 0</a> on page 12
Additional register ID_AA64ZFR0_EL1, SVE Feature ID register 0 description added	<a href="#">3.3 ID_AA64ZFR0_EL1, SVE Feature ID register 0</a> on page 16

**Table A-3: Differences between 0000-02 and 0001-03**

Change	Location
First early access release for r0p1	-
Editorial revisions	Throughout document
Updated ID_AA64ISAR0_EL1 register	<a href="#">3.2 ID_AA64ISAR0_EL1, AArch64 Instruction Set Attribute Register 0</a> on page 12
Updated ID_AA64ZFR0_EL1 register	<a href="#">3.3 ID_AA64ZFR0_EL1, SVE Feature ID register 0</a> on page 16

**Table A-4: Differences between 0001-03 and 0001-04**

Change	Location
Second early access release for r0p1	-
Editorial revisions	Throughout document
Updated ID_AA64ISAR0_EL1 register	<a href="#">3.2 ID_AA64ISAR0_EL1, AArch64 Instruction Set Attribute Register 0</a> on page 12
Updated ID_AA64ZFR0_EL1 register	<a href="#">3.3 ID_AA64ZFR0_EL1, SVE Feature ID register 0</a> on page 16

**Table A-5: Differences between 0001-04 and 0002-05**

Change	Location
First release for r0p2	-
Editorial revisions	Throughout document
Updated product revision	Throughout document
Updated ID_AA64ISAR0_EL1 register	<a href="#">3.2 ID_AA64ISAR0_EL1, AArch64 Instruction Set Attribute Register 0</a> on page 12
Updated ID_AA64ZFR0_EL1 register	<a href="#">3.3 ID_AA64ZFR0_EL1, SVE Feature ID register 0</a> on page 16