Arm® Cortex®-A35 Processor Cryptographic Extension

Revision: r1p0

Technical Reference Manual

arm

100237_0100_00_en

Arm® Cortex®-A35 Processor Cryptographic Extension

Technical Reference Manual

Release Information

Document History

Issue	Date	Confidentiality	Change
0000-00	04 December 2015	Confidential	First release for r0p0
0001-00	18 March 2016	Confidential	First release for r0p1
0002-00	04 March 2017	Non-Confidential	First release for r0p2
0100-00	28 February 2019	Non-Confidential	First release for r1p0

Non-Confidential Proprietary Notice

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®-A35 Processor Cryptographic Extension Technical Reference Manual

Preface

This preface introduces the *Arm® Cortex®-A35 Processor Cryptographic Extension Technical Reference Manual*.

It contains the following:

About this book

A technical reference document that describes the optional cryptographic features of the Cortex®-A35 processor. It includes descriptions of the registers used by the Cryptographic Extension.

Product revision status

The r*m*p*n* identifier indicates the revision status of the product described in this book, for example, r*1*p*2*, where:

r*m* Identifies the major revision of the product, for example, r1.

p*n* Identifies the minor revision or modification status of the product, for example, p2.

Intended audience

This manual is written for system designers, system integrators, and programmers who are designing or programming a System-on-Chip (SoC) that uses the Cortex®-A35 processor 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-A35 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® 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.

`mono`space
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.

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.

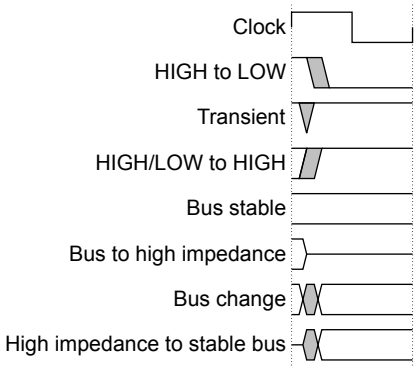

Figure 1 Key to timing diagram conventions

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 denotes an active-LOW signal.

Additional reading

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

Arm publications

- *Arm® Cortex®-A35 Processor Technical Reference Manual* (100236).
- *Arm® Cortex®-A35 Configuration and Sign-off Guide* (100239).
- *Arm® Cortex®-A35 Processor Integration Manual* (100240).
- *Arm® Cortex®-A35 Processor Advanced SIMD and Floating-point Support Technical Reference Manual* (100238).
- *Arm® Architecture Reference Manual Armv8, for Armv8-A architecture profile* (DDI 0487).

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-A35 Processor Cryptographic Extension Technical Reference Manual*.
- The number 100237_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
Functional Description

This chapter describes the Cortex-A35 Cryptographic Extension.

It contains the following sections:

1.1 About the Cryptographic Extension

The Cortex-A35 processor Cryptographic Extension supports the Armv8 Cryptographic Extensions.

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 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-A35 processor and Advanced SIMD and floating-point support licenses.

1.2 Revisions

This section describes the differences in functionality between product revisions.

r0p0 First release.

r0p1 There are no functional changes in this revision.

r0p2 There are no functional changes in this revision.

r1p0 There are no functional changes in this revision.

Chapter 2
Register Descriptions

This chapter describes the Cryptographic Extension registers.

It contains the following sections:

2.1 Identifying the cryptographic instructions implemented

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

The three registers are:

- ID_AA64ISAR0_EL1 in the AArch64 execution state.
- ID_ISAR5_EL1 in the AArch64 execution state.
- ID_ISAR5 in the AArch32 execution state.

2.2 Disabling the Cryptographic Extension

To disable the Cryptographic Extension for each individual core, assert the corresponding bit of the **CRYPTODISABLE** input signal. This signal is only sampled during reset of the core.

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 2-16 indicate that the Cryptographic Extension is not implemented.

2.3 Register summary

The processor has three 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-A35 processor Cryptographic Extension.

Table 2-1 Cryptographic Extension register summary

Name	Execution state	Description
ID_AA64ISAR0_EL1	AArch64	See *2.4 AArch64 Instruction Set Attribute Register 0, EL1* on page 2-17.
ID_ISAR5_EL1	AArch64	See *2.5 AArch32 Instruction Set Attribute Register 5, EL1* on page 2-19.
ID_ISAR5	AArch32	See *2.6 Instruction Set Attribute Register 5* on page 2-21.

2.4 AArch64 Instruction Set Attribute Register 0, EL1

The ID_AA64ISAR0_EL1 characteristics are:

Purpose

Provides information about the instructions implemented in AArch64 state, including the instructions provided by the Cryptographic Extension.

Note

The optional Cryptographic Extension is not included in the base product of the processor. Arm requires licensees to have contractual rights to obtain the Cryptographic Extension.

Usage constraints

This register is accessible as follows:

EL0	EL1 (NS)	EL1 (S)	EL2	EL3 (SCR.NS = 1)	EL3 (SCR.NS = 0)
-	RO	RO	RO	RO	RO

Configurations

ID_AA64ISAR0_EL1 is architecturally mapped to external register ID_AA64ISAR0.

Attributes

ID_AA64ISAR0_EL1 is a 64-bit register.

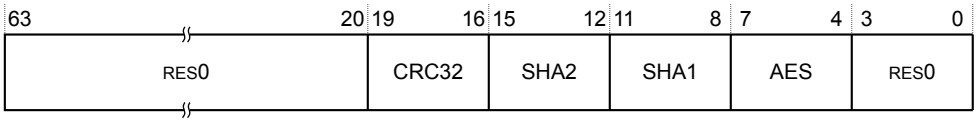

Figure 2-1 ID_AA64ISAR0_EL1 bit assignments

[63:20]

Reserved, RES0.

CRC32, [19:16]

Indicates whether CRC32 instructions are implemented. The possible values are:

`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 processor implementation does not include the Cryptographic Extension.

`0x1` `SHA256H`, `SHA256H2`, `SHA256U0`, and `SHA256U1` implemented. This is the value if the processor implementation includes the Cryptographic Extension.

SHA1, [11:8]

Indicates whether SHA1 instructions are implemented. The possible values are:

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

`0x1` `SHA1C`, `SHA1P`, `SHA1M`, `SHA1SU0`, and `SHA1SU1` implemented. This is the value if the processor 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 processor implementation does not include the Cryptographic Extension.

`0x2` `AESE`, `AESD`, `AESMC`, and `AESIMC` implemented, plus `PMULL` and `PMULL2` instructions operating on 64-bit data. This is the value if the processor implementation includes the Cryptographic Extension.

[3:0]

Reserved, RES0.

To access the ID_AA64ISAR0_EL1:

```
MRS <Xt>, ID_AA64ISAR0_EL1 ; Read ID_AA64ISAR0_EL1 into Xt
```

ID_AA64ISAR0_EL1[31:0] can be accessed through the internal memory-mapped interface and the external debug interface, offset `0xD30`.

Register access is encoded as follows:

Table 2-2 ID_AA64ISAR0_EL1 access encoding

op0	op1	CRn	CRm	op2
11	000	0000	0110	000

2.5 AArch32 Instruction Set Attribute Register 5, EL1

The ID_ISAR5_EL1 characteristics are:

Purpose

Provides information about the instructions implemented in AArch32 state, including the instructions provided by the optional Cryptographic Extension.

Note

The optional Cryptographic Extension is not included in the base product of the processor. Arm requires licensees to have contractual rights to obtain the Cryptographic Extension.

Usage constraints

This register is accessible as follows:

EL0	EL1 (NS)	EL1 (S)	EL2	EL3 (SCR.NS = 1)	EL3 (SCR.NS = 0)
-	RO	RO	RO	RO	RO

Configurations

ID_ISAR5_EL1 is architecturally mapped to AArch32 register ID_ISAR5. See *2.6 Instruction Set Attribute Register 5* on page 2-21.

Attributes

ID_ISAR5_EL1 is a 32-bit register.

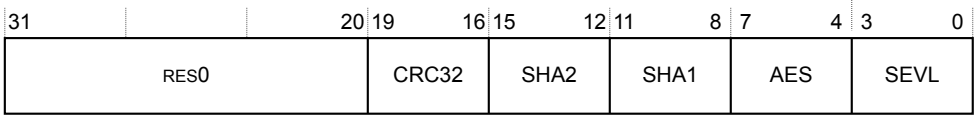

Figure 2-2 ID_ISAR5_EL1 bit assignments

[31:20]

Reserved, RES0.

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 Extensions are not implemented or are 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 Extensions are not implemented or are 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 Extensions are not implemented or are 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` `SEVL` implemented to send event local.

To access the ID_ISAR5_EL1:

```
MRS <Xt>, ID_ISAR5_EL1 ; Read ID_ISAR5_EL1 into Xt
```

Register access is encoded as follows:

Table 2-3 ID_ISAR5_EL1 access encoding

op0	op1	CRn	CRm	op2
11	000	0000	0010	101

2.6 Instruction Set Attribute Register 5

The ID_ISAR5 characteristics are:

Purpose

Provides information about the instructions implemented in AArch32 state, including the instructions provided by the optional Cryptographic Extension.

Note

The optional Cryptographic Extension is not included in the base product of the processor. Arm requires licensees to have contractual rights to obtain the Cryptographic Extension.

Usage constraints

This register is accessible as follows:

EL0 (NS)	EL0 (S)	EL1 (NS)	EL1 (S)	EL2	EL3 (SCR.NS = 1)	EL3 (SCR.NS = 0)
-	-	RO	RO	RO	RO	RO

The ID_ISAR5 must be interpreted with ID_ISAR0, ID_ISAR1, ID_ISAR2, ID_ISAR3, and ID_ISAR4.

Configurations

ID_ISAR5 is architecturally mapped to AArch64 register ID_ISAR5_EL1. See *2.5 AArch32 Instruction Set Attribute Register 5, EL1* on page 2-19.

There is one copy of this register that is used in both Secure and Non-secure states.

Attributes

ID_ISAR5 is a 32-bit register.

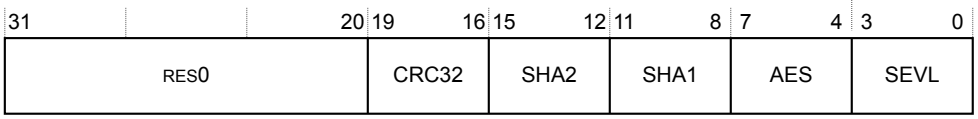

Figure 2-3 ID_ISAR5 bit assignments

[31:20]

Reserved, RES0.

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 extensions are not implemented or are 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 extensions are not implemented or are 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 extensions are not implemented or are disabled.

`0x2` `AESE`, `AESD`, `AESMC` and `AESIMC`, plus `PMULL` and `PMULL2` instructions operating on 64-bit data.

SEVL, [3:0]

Indicates whether the `SEVL` instruction is implemented. The value is:

`0x1` `SEVL` implemented to send event local.

To access ID_ISAR5:

```
MRC p15, 0, <Rt>, c0, c2, 5; Read ID_ISAR5 into Rt
```

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-24.

A.1 Revisions

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

Table A-1 Issue 0000-00

Change	Location	Affects
First release for r0p0	-	-

Table A-2 Issue 0001-00

Change	Location	Affects
First release for r0p1	On front page and in document history table.	r0p1
Product name udpated from the Mercury processor to the Cortex-A35 processor.	Everywhere the product name is used.	All versions

Table A-3 Issue 0002-00

Change	Location	Affects
First release fo r0p2	On front page and in document history table.	r0p2

Table A-4 Issue 0100-00

Change	Location	Affects
First release fo r1p0	On front page and in document history table.	r1p0