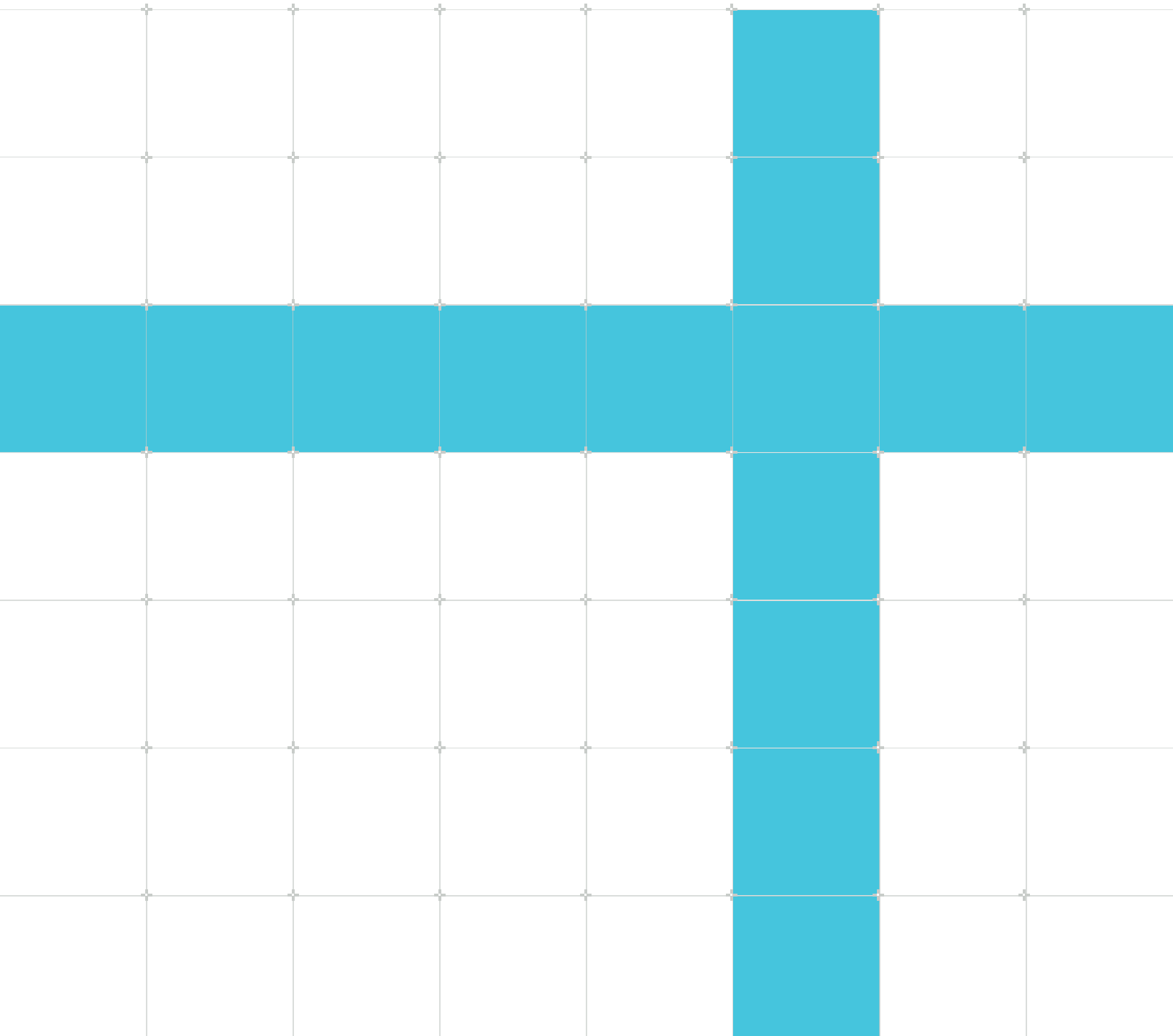
arm

Learn the architecture - Understanding the Armv8.x and Armv9.x extensions

Version 2.1

Non-Confidential
Copyright © 2019, 2021 Arm Limited (or its affiliates).
All rights reserved.

Issue 02
102378_0201_02_en



Learn the architecture - Understanding the Armv8.x and Armv9.x extensions

Copyright © 2019, 2021 Arm Limited (or its affiliates). All rights reserved.

Release information

Document history

Issue	Date	Confidentiality	Change
0100-01	1 April 2019	Non-Confidential	First release
0200-01	30 March 2021	Non-Confidential	Updates for Armv9-A
0201-02	8 September 2021	Non-Confidential	Extensions and features update

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, 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.

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 https://www.arm.com/company/policies/trademarks.

Copyright © 2019, 2021 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.

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.

Copyright © 2019, 2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

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.

We believe that this document contains no offensive language. To report offensive language in this document, email terms@arm.com.

Copyright © 2019, 2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

Contents

Copyright © 2019, 2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

1. Overview

Additions to the Arm architecture are provided as version increments called extensions. Extensions allow us to release new features regularly in response to the needs of our partners without making major changes to the main architecture.

Arm releases a new extension every year. Cortex CPUs, which are Arms implementations of the architecture, use the latest extension depending on when they are released.

This guide explains the extensions to the Arm architecture and provides guidance on how to read and use them.

At the end of this guide, you can check your knowledge. You will have learned the following:

- The naming scheme used to identify extensions.
- Which features are available in different extensions.
- How to determine which features of extension an Arm Cortex CPU implementation supports.

Copyright © 2019, 2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

2. What do Armv8.x-A and Armv9.x mean?

There are different versions of the Arm architecture. These different versions are usually shown as ArmvX, in which X is the version number. For example, Armv8-A means version 8 of the Arm A-profile architecture and Armv9-A means version 9 of the A-profile architecture. A version like Armv8-A is a major release of the architecture.

However, there are also minor versions that are added to a major release. These minor versions are called .x extensions. For example, Armv8.1-A means version 8 of the A-profile architecture, extended by the .1 extension.

Copyright © 2019, 2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

3. Why do we need the .x extensions?

Development of major versions of the Arm architecture can take many years. For example, Armv7-A was released in 2007 and Armv8-A followed six years later, in 2013. Because the architecture needs to evolve between major versions to add new features, minor versions, the .x extensions, are added.

Since the release of Armv8-A, the process of adding to the architecture between major versions has been formalized. There is now an annual release of a .x extension. Beginning with Armv8.0-A, the base specification, the Armv8.1-A extension was added in 2015, then the Armv8.2-A extension was added in 2016. Each .x extension builds on the last. For example, Armv8.2-A includes all the features of Armv8.1-A, and adds new features. This process continues with Armv9-A.

Extension naming

Features in the architecture have an identifier, in the form FEAT_<name>. For example:

- FEAT_SHA256 identifies support for the SHA256 operations added in Armv8.2-A.
- FEAT_BRBE identifies support for the Branch-Record Buffer extension added in Armv9.2-A.

You can find a full list of feature names on the Arm Developer site.

The Arm architecture documentation uses these identifiers extensively. For example, here is the description of the AMCFGR register:

```
    This register is present only when FEAT_AMUv1 is implemented. Otherwise,
    direct accesses to AMCFGR are UNDEFINED.
```

The Arm Architecture Reference Manual provides information on which features are optional or mandatory in different versions of the Armv8.x-A and Armv9.x-A architectures.

Copyright © 2019, 2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

4. Processor implementation

Each .x extension includes a set of features, some mandatory and some optional. A processor implements a .x extension if it implements all the mandatory features of that extension number, and the mandatory features from all previous extensions.

For example, a processor that is described as implementing Armv8.2-A must implement all the mandatory features from the following architecture releases and extensions:

- Armv8.0-A - the base specification and original release
- Armv8.1-A - the previous extension
- Armv8.2-A - the new extension

Similarly, a processor that is described as implementing Armv9.1-A must implement all the mandatory features from the following architecture releases and extensions:

- Armv9.0-A - the base specification and original release
- Armv9.1-A - the new extension

A feature might originally be optional, but later become mandatory. For example, the Dot Product instructions were optional in all extensions from Armv8.0-A to Armv8.3-A, but became mandatory in Armv8.4-A. Similarly, a feature might have been optional in Armv8-A but mandatory in Armv9-A.

An Armv8.x-A processor can implement any features from the next .x extension. However, it cannot implement features from any later .x extension.

For example, a processor described as implementing Armv8.1-A:

- Must implement all the mandatory features of Armv8.0-A and Armv8.1-A
- Is permitted to implement some features from Armv8.2-A
- Is not permitted to implement features from Armv8.3-A and later extensions

Copyright © 2019, 2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

5. Armv8.x and Armv9.x extensions and features

In this section of the guide, we summarize the new features that were added in each of the Armv8.x-A and Armv9.x-A extensions. We do not provide a complete list, but we include the most important features. Notice that some features are limited to the AArch64 state, and others are available in both the AArch32 and AArch64 states.

AArch32 is a 32-bit Execution state that is supported in all versions of Arm architecture before Armv8-A. AArch64 is a 64-bit Execution state and is supported only in the Armv8-A architecture.

Armv8.1-A

- Atomic memory access instructions (AArch64)
- Limited Order regions (AArch64)
- Increased Virtual Machine Identifier (VMID) size, and Virtualization Host Extensions (AArch64)
- Privileged Access Never (PAN) (AArch32 and AArch64)

Armv8.2-A

- Support for 52-bit addresses (AArch64)
- The ability for PEs to share Translation Lookaside Buffer (TLB) entries (AArch32 and AArch64)
- FP16 data processing instructions (AArch32 and AArch64)
- Statistical profiling (AArch64)
- Reliability Availability Serviceabilty (RAS) support becomes mandatory (AArch32 and AArch64)

Armv8.3-A

- Pointer authentication (AArch64)
- Nested virtualization (AArch64)
- Advanced Single Instruction Multiple Data (SIMD) complex number support (AArch32 and AArch64)
- Improved JavaScript data type conversion support (AArch32 and AArch64)
- A change to the memory consistency model (AArch64)
- ID mechanism support for larger system-visible caches (AArch32 and AArch64)

Armv8.4-A

- Secure virtualization (AArch64)
- Nested virtualization enhancements (AArch64)

Copyright © 2019, 2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

- Small translation table support (AArch64)
- Relaxed alignment restrictions (AArch32 and AArch64)
- Memory Partitioning and Monitoring (MPAM) (AArch32 and AArch64)
- Additional crypto support (AArch32 and AArch64)
- Generic counter scaling (AArch32 and AArch64)
- Instructions to accelerate SHA

Armv8.5-A and Armv9.0-A

- Memory Tagging (AArch64)
- Branch Target Identification (AArch64)
- Random Number Generator instructions (AArch64)
- Cache Clean to Point of Deep Persistence (AArch64)

Armv8.6-A and Armv9.1-A

- General Matrix Multiply (GEMM) instructions (AArch64)
- Fine grained traps for virtualization (AArch64)
- High precision Generic Timer
- Data Gathering Hint (AArch64)

Armv8.7-A and Armv9.2-A

- Enhanced support for PCIe hot plug (AArch64)
- Atomic 64-byte load and stores to accelerators (AArch64)
- Wait For Instruction (WFI) and Wait For Event (WFE) with timeout (AArch64)
- Branch-Record recording (Armv9.2 only)

Armv8.8-A and Armv9.3-A

- Non-maskable interrupts (AArch64)
- Instructions to optimize `memcpy()` and `memset()` style operations (AArch64)
- Enhancements to PAC (AArch64) Hinted conditional branches (AArch64)

Copyright © 2019, 2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

6. Which extension does my processor implement?

The Arm architecture includes a set of feature registers that report the features supported by the processor. For each new feature added by a .x extension, even the optional features, a field in these feature registers reports whether it is supported or not.

For example, ID_AA64MMFR2_EL1.AT tells you whether there is support for the relaxed alignment requirements in Armv8.4-A. There is no field that reports whether a processor is Armv8.1-A. Instead, software reads the feature fields for the mandatory 8.1-A features, and if they all present, the processor is compliant with Armv8.1-A.

Copyright © 2019, 2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

7. Armv8.x-A and the SBSA

The Server Base System Architecture (SBSA), provides hardware requirements for servers. The SBSA ensures that operating systems, hypervisors and firmware operate correctly. For servers, where a degree of standardization is important, the SBSA includes rules on which extensions to the architecture must be implemented.

The following table summarizes the SBSA requirements that relate to the Armv8.x-A extensions:

Version	Feature	SBSA Level 3	Level 4	Level 5
Armv8.0-A	Advanced SIMD Crypto instructions CRC 4KB and 64KB granule 16-bit ASI EL2 and EL3 AArch64 at all Exception levels At least six PMU counters At least six breakpoints and four synchronous watchpoints	Mandatory Mandatory (subject to export restrictions) Mandatory Mandatory Mandatory Mandatory Mandatory Mandatory Mandatory		
Armv8.1-A	16-bit VMIDs Virtualization Host Extension		Mandatory Mandatory	
Armv8.2-A	RAS Persistent memory		Mandatory (at least minimal implementation) Optional (with restrictions)	
Armv8.3-A	Nested virtualization Pointer authentication	Optional (with restrictions)	Mandatory	Optional (with restrictions)
Armv8.4-A	Stage 2 type overrides Enhanced nested virtualization Activity monitors MPAM SHA3 and SHA512 Generic counter scaling			Mandatory Mandatory Mandatory Optional (with restrictions) Mandatory (subject to export restrictions) Mandatory

Copyright © 2019, 2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

8. Check your knowledge

Q: What are the major version and .x extension in Armv8.3?

8 is the version, and 3 is the .x extension.

Q: Secure virtualization was added in Armv8.4-A. Would an Armv8.1-A processor be allowed to implement it?

No. An Armv8.1 processor must implement the mandatory features of Armv8.1 and may implement features from Armv8.2-A. But it is not allowed to implement features from Armv8.3, Armv8.4, or later versions, unless a special concession has been made.

Q: Is this sentence true or false? Only optional features have fields to report their presence.

False. Mandatory and optional features have fields to report their presence.

Q: Which major version and extension of the Arm architecture does the Cortex-A55 implement?

Armv8.2-A.

Q: Which level or levels of the SBSA require support for 16-bit VMIDs?

Level 3 and above.

Copyright © 2019, 2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

9. Related information

To learn more about the following features referenced in this guide, refer to:

- Security guides:
  - Introduction to security
  - TrustZone for AArch64
  - Providing protection for complex software
- Armv8-A Virtualization

Here are some resources related to material in this guide:

- Arm architecture and reference manuals (for information on Armv8.1-A, Armv8.2-A, Armv8.3-A, Armv8.4-A)
- Arm community (ask development questions, and find articles and blogs on specific topics from Arm experts)

Here are some resources related to topics in this guide:

- Armv8-x.A and the SBSA:
  - Server Base System Architecture (SBSA)

Copyright © 2019, 2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential

10. Next steps

Every year, Arm releases extensions to its main architecture providing new features in support of partner needs. In this guide, we explained the extensions to the Armv8.x architecture, described how to read and use the extensions, and outlined some of the features that the extensions support.

After reviewing this guide, you should understand how the extensions are expressed, which features are available in which extensions, and how to determine which features an Arm Cortex CPU implementation supports.

To keep learning about the Armv8-A and Armv9-A architectures, see more in our series of guides.

Copyright © 2019, 2021 Arm Limited (or its affiliates). All rights reserved.
Non-Confidential