Arm® RME System Architecture Compliance Suite

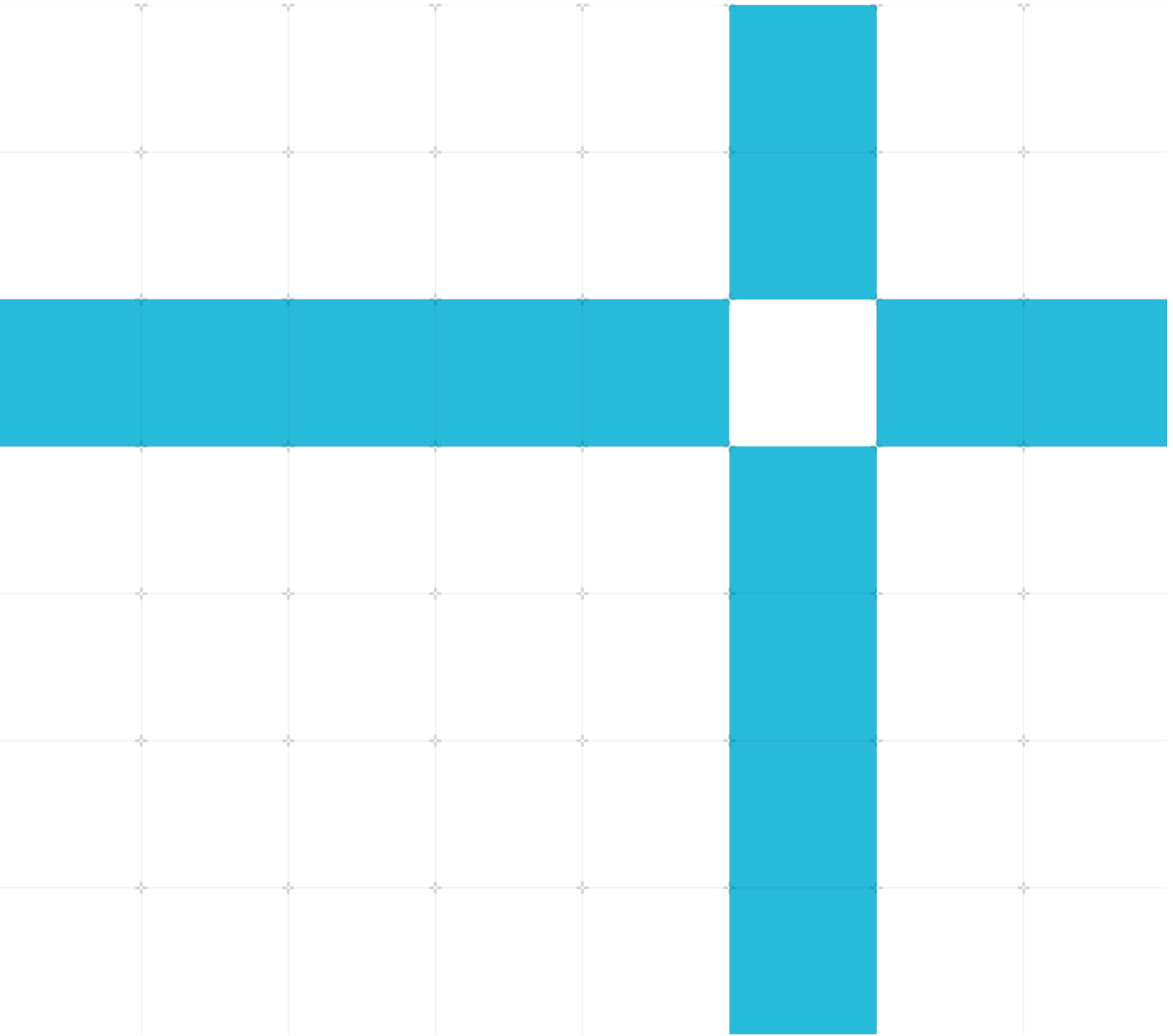
Version: 4.0

Scenario Document

Non-Confidential

Issue 07

PJDOC-1505342170-664675



RME System Architecture Compliance Suite

Scenario Document

Release information

Document history

Issue	Date	Confidentiality	Change
01	25-Apr-2023	Confidential	First release for v0.5
02	25-Aug-2023	Confidential	First release for v0.6
03	06-Nov-2023	Non-Confidential	First release for v0.7
04	13-Dec-2023	Non-Confidential	First release for v1.0
05	01-Dec-2024	Non-Confidential	First release for v2.0
06	21-Jan-2025	Non-Confidential	First release for v3.0
07	13-Jun-2025	Non-Confidential	First release for v4.0

Non-Confidential Proprietary Notice

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.

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 for a final product, that is for a developed product.

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on Arm® RME System Architecture Compliance Suite Scenario Document, 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.

Contents

1. Introduction

1.1 Intended audience

This document is for engineers who are verifying an implementation of Arm® RME enabled System.

1.2 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.

Typographic conventions

Convention	Use
italic	Citations.
bold	Highlights interface elements, such as menu 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
<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**.

1.3 Useful resources

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

- Arm Non-Confidential documents are available on developer.arm.com/documentation. Each document link in the tables below provides direct access to the online version of the document.
- Arm Confidential documents are available to licensees only through the product package.

Arm products	Document ID	Confidentiality
Arm® Realm Management Extension (RME) System Architecture	DEN0129H	Non-Confidential
Arm® System Memory Management Unit Architecture Specification	IHI0070	Non-Confidential

Arm architecture and specifications	Document ID	Confidentiality
Arm® Architecture Reference Manual for A-profile architecture	DDI0487	Non-Confidential
Arm® Generic Interrupt Controller Architecture Specification for GIC architecture version 3.0 and version 4.0	IHI0069C	Non-Confidential

Non-Arm resources	Document ID	Organization

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. About the Arm® RME system ACS

This chapter introduces you to the Arm® RME system Architecture Compliance Suite.

2.1. Abbreviations

The following table lists the abbreviations used in this document.

Table 2-1: Abbreviations and expansions

Abbreviation	Expansion
ACS	Architecture Compliance Suite
RNVS	Root Non-volatile Storage
PAL	Platform Abstraction Layer
MSD	Monitor Security Domain
PAS	Physical Address Space
SMEM	Shielded memory
RT	Root
RL	Realm
NS	Non-Secure
PA_(X)	(X) is access PAS encoding in S/NS/RT/RL
DA	Device Assignment
DPT	Device Permission Table
TEE	Trusted Execution Environment
TDISP	TEE Device Interface Security Protocol
RMSD	Realm Management Security Domain
IDE	Integrity and Data Encryption
KM	Key Management
TDI	TEE Device Interface
MEC	Memory Encryption Contexts
MECID	Memory Encryption Contexts Identifier
CMO	Cache Maintenance Operation
PoC	Point of Coherence
PoPA	Point of Physical Aliasing
PoE	Point of Encryption

2.2. Scope of this document

This document describes the verification scenarios and the strategy that is followed for creating Architecture Compliance Suite (ACS) tests for Realm Management Extension (RME) System Architecture. The scenarios are referred from the Arm® Realm Management Extension (RME) System Architecture Specification.

2.3. Introduction to RME system ACS

RME System Architecture is an extension to the Armv9 A-profile architecture. It adds the following features:

- Two additional Security states, Root and Realm.
- Two additional physical address spaces, Root and Realm.
- The ability to dynamically transition memory granules between physical address spaces.
- Granule Protection Check mechanism.

The RME architecture defines the set of hardware features and properties that are required to comply with the Arm ***Confidential Compute Architecture (CCA***). Implementations compliant with the RME System architecture must conform to the behavior described in the specification.
The Architecture Compliance Suite (ACS) is a set of examples of the specified invariant behaviors. Use this suite to verify that these behaviors are implemented correctly in your system.

3. Test scenarios

This chapter describes the execution flow control used for RME System ACS.

3.1. PE

Scenario ID: 1.1

Rules:
RBJVZS: An access to a Resource is associated with an Access PAS in accordance with the PAS Access Table.
RGDVSZ: A PA of an access to a memory-mapped peripheral is associated with a PAS until reaching the PAS filter assigned to protect the peripheral.
RDVPGT: A private PAS filter allows access to a register only if the Access PAS matches a Resource PAS that the register is associated with.
Scenario: All protected memory regions are accessible only when resource PAS and access PAS are same.

Note: If regions are protected only by GPC, then verify with GPC enabled.

Observation:
Accesses with same resource PAS and access PAS are successful otherwise generates fault.

Scenario ID: 1.2

Rules: RWFQKD: A PA that targets memory that can be cached is associated with a PAS until reaching the PoPA.
RFRMJJ: Where a PA is associated with a PAS, any PA compared operation includes the PAS.

Scenario ID: 1.2.a

A location PA1 is marked as “All Access Permitted” in GPT.
Cacheable store to PA1_NS is not visible to PA1_RL, PA1_RT, PA1_S.

Scenario ID: 1.2.b

MP setup to validate snoop filter considers PAS:
PA1 is marked as Shareable in both PE0 & PE1.
PE0: PA1 marked as Root PAS.
PE1: PA1 marked as Realm PAS.
PE1 must not generate snoop access to PE0. As a result, update to PE0.

Scenario ID: 1.3

Rule: The first part of the rule below is validated in this scenario.
RMLFBL: External memory that is assigned to Secure PAS, Realm PAS, or Root PAS must be encrypted using a method that provides at least the following:

- A different encryption context for each PAS.
- A different address tweak for each encryption data block, such as a 128-bit memory block.
- If cryptographic memory integrity is not supported, an encryption mode ensures bit diffusion over an encryption data block.

Scenario:
Data is encrypted when written in memory/any shared cache beyond PoPA.
PA1 is marked as All Access Permitted.
Store DATA1 in PA1_RT; CMO till PoPA (for all PAS); Read using PA_S return DATA2; READ using PA_RL return DATA3; READ using PA_NS return DATA4.

Observation:
DATA1! = DATA2! = DATA3! = DATA 4
Note: The third point of this rule is not validated in ACS.

Scenario ID: 1.4

Rule: The second part of the rule below is validated in this scenario.
RMLFBL: External memory that is assigned to Secure PAS, Realm PAS, or Root PAS must be encrypted using a method that provides at least the following:

- A different encryption context for each PAS.
- A different address tweak for each encryption data block, such as a 128-bit memory block.
- If cryptographic memory integrity is not supported, an encryption mode ensures bit diffusion over an encryption data block.

Scenario:
Data is encrypted with a different tweak in each 128-bit of data block.
Store DATA1 in PA1_S and (PA1_S + 16).
CMO to PoPA using S and NS PAS.
Read PA1_NS returns DATA2; Read PA1_NS+ 16 returns DATA3.

Observation:
DATA2 and DATA3 are different.

Scenario ID: 1.5

Rules:

RNXJLB: On an RME system reset MSD SMEM is either immediately assigned to the Root PAS or scrubbed and is available for access by the PE boot ROM as soon as it starts executing.
RCSSDG: MSD SMEM is in the Root PAS.
Scenario:
MSD SMEM is in ROOT PAS. Access MSD SMEM with S, NS, RT & RL access PAS.
Observation:
Only RT accesses are successful.

Scenario ID: 1.6

Rules:
RCMMCZ: RMSD SMEM is in the Realm PAS.
RZQQSQ: SMEM that can be dynamically assigned to the Realm PAS, or the Secure PAS is either immediately assigned to the Root PAS or scrubbed on an RME system reset.

Scenario:
Verify that Realm SMEM is in realm PAS (if Realm SMEM is defined statically).
Observation:
Root, Secure and Non-secure access to Realm SMEM returns error.

Scenario ID: 1.7

Rule:
RGSRPS: All A-profile application PEs in the system implement the Realm Management Extension (RME).

Scenario:
All application PEs implement RME.
Observation:
Read of ID_AA64PFR0_EL1.RME returns >=1 for all PEs.

Scenario ID: 1.8

Rules:
RXTSXB: An RME coherent interconnect supports cache maintenance operations to the PoPA.
RLCXDB: Completion of a PoPA CMO for a given PA guarantees that:

- Any dirty cached or transient state associated with the PA before the PoPA has been cleaned to after the PoPA.
- Any cached or transient state associated with the PA before the PoPA has been invalidated.

Scenario:
Coherent interconnect supports CMO to PoPA. CMO to PoPA cleans dirty copy till PoPA and invalidates all cached copies.
Stimulus:

- PA1 is marked as All Access Permitted. PA1 is initialized with random data.
- VA1 is mapped to PA1 as secure PAS in MMU with cacheable attribute.
- VA2 is mapped to PA1 as nonsecure in MMU with cacheable attribute.
- Read VA1 returns data1.
- Read VA2 returns data2.
- Store data3 in VA1.
- Issue CMO to PoPA for PA1 secure and nonsecure.

Observation:
Read of VA2 must return data4 (! = data2) (ciphertext of data3).

Scenario ID: 1.9

Rules:
RFXQCD: A PoPA CMO applies to any cache before the PoPA, including system caches that are located beyond the Point of Coherency.
RQBNJF: A PoPA CMO applies to any cached copy in the system with the specified {PAS, PA} regardless of both:

- The shareability domain it was cached with.
- Whether the system supports a single or multiple Outer Shareable shareability domains.

Scenario:
repeat 1.8 with Non-Cacheable memory.

Scenario ID: 1.10

Rule:
RJRJSQ: An RME coherent interconnect complies with a Distributed Virtual Memory (DVM) version that supports Realm Translation Regimes and TLB Invalidate by PA operations.

Scenario:
Check interconnect supports TLBI PA operation by changing GPT entry.
Stimulus:

- Map VA1 to PA1 as secure memory both in MMU and GPT.
- Access VA1.
- Change PA1 to non-secure using Undelegated algo.
    - Issue TLBI PA as a part of undelegated algorithm.

Observation:
Access to VA1 will generate fault.

Internal Note: This scenario will be repeated in SMMU environment and default validation will be in MP environment.

Scenario ID: 1.11

Rule:
RQDPVN: Any PAS other than the Non-secure PAS must have encryption enabled.

Scenario:
Secure, Realm and Root PAS must have encryption enabled.
Stimulus:

- PA1 marked as all access permitted.
- PA1 is mapped as VA1_S, VA2_NS, VA3_RL, VA4_RT.
- Store data1 using VA2_NS.

Observation:
Read of VA1_S, VA3_RL, VA4_RT must return unique values (! =data1).

Scenario ID: 1.12

Rule:
RVSMPS: The decision to enable encryption for the Non-secure PAS is either hardwired or defined at boot and immutable once set.

Scenario: NSencryption(enable=1).
Once enabled then we cannot disable by calling NSencryption(enable=0).
Stimulus:

- Store data1 in PA1_NS.
- Read PA1_NS will return data1.
- CMO to PoPA for PA1.
- Enable NS encryption.
- CMO to PoPA for PA1.
- Read of PA1_NS will return data2 which is not same as data1.
- Disable NS encryption.
- CMO to PoPA.

Observation: Read of PA1 still returns data2.

Scenario ID: 1.13

Rule:
RJYMQD: Allocation and protection of the address range assigned to an MTE carve-out are controlled by either SSD or MSD.

Scenario:
MTE carve-out region is not accessible by NS/S/RL accesses.

Observation:
NS, S and RL accesses to MTE carve-out region generates fault (fault might not be generated).

Scenario ID: 1.14

Rule:
RSQMWT: Application PEs in an RME system do not have architectural differences unless this is explicitly permitted by this specification.

Scenario: Verify application PEs in an RME system do not have any architectural differences.
Below registers list will be checked:
AARCH32 registers are checked when ARCH32 are supported at EL0.

Register Name	Register mask	Dependency
CCSIDR_EL1	0xFFFFFF8	-
ID_AA64PFR0_EL1	-	
ID_AA64PFR1_EL1	-	
ID_AA64DFR0_EL1	-	
ID_AA64DFR1_EL1	-	
ID_AA64MMFR0_EL1	0xF	
ID_AA64MMFR1_EL1	-	
CTR_EL0	0xC000	
ID_AA64ISAR0_EL1	-	
ID_AA64ISAR1_EL1	-	-
MPIDR_EL1	0xFF3FFFFFFF	-
MIDR_EL1	0x00F0FFFF	-
ID_DFR0_EL1	-	AARCH32
ID_ISAR0_EL1	-	AARCH32
ID_ISAR1_EL1	-	AARCH32
ID_ISAR2_EL1	-	AARCH32
ID_ISAR3_EL1	-	AARCH32
ID_ISAR4_EL1	-	AARCH32
ID_ISAR5_EL1	-	AARCH32
ID_MMFR0_EL1	-	AARCH32
ID_MMFR1_EL1	-	AARCH32
ID_MMFR2_EL1	-	AARCH32
ID_MMFR3_EL1	-	AARCH32
ID_MMFR4_EL1	-	AARCH32

Register Name	Register mask	Dependency
ID_PFR0_EL1	-	AARCH32
ID_PFR1_EL1	-	AARCH32
MVFR0_EL1	-	AARCH32
MVFR1_EL1	-	AARCH32
MVFR2_EL1	-	AARCH32
PMCEID0_EL0	-	PMUV3
PMCEID1_EL0	-	PMUV3
PMCR_EL0	0xFFFF	PMUV3
PMBIDR_EL1	-	SPE
PMSIDR_EL1	-	SPE
ERRIDR_EL1	-	RAS
ERR0FR_EL1	-	RAS
ERR1FR_EL1	-	RAS
ERR2FR_EL1	-	RAS
ERR3FR_EL1	-	RAS
LORID_EL1	-	LOR

Scenario ID: 1.15

Rule:
RMLJVR: On an exit from a low power state in which system context is preserved, power control guarantees that MSD state is fully preserved. If MSD state is not preserved, power control applies an RME system reset.

Scenario ID: 1.15.a

Check MSD state is preserved after exit from low power state. PE will use WFI to enter a low power state.

Scenario ID: 1.15.b

Check MSD state is preserved after exit from PE Suspend.

Note:

1. The test will check granule protection control registers.
2. Sanity checks the content of GPT tables.
3. SMMU tables and controls are preserved.
4. MPE is enabled.

Scenario ID: 1.16

Rule:
RZNLSZ: Save/Restore operations for MSD state can only be done by MSD or a Trusted subsystem and use on-chip storage that is not accessible from Realm PAS, Secure PAS or Non-secure PAS.

Scenario:
MSD state save restore location not accessible via S/NS/RL accesses.

Scenario ID: 1.17

Rule:
RQCHPW: The system supports a method for permanently blocking write access from application PEs to all RNVS parameters.

Scenario:
RNVS programming functions (memory mapped) can only be accessed from Root PAS.

Observation:
Non-Root access to RNVS programming functions generate faults.
Note: Review PAL function after implementation. We can test mailbox is not accessible from non-Root PAS.

Scenario ID: 1.18

Rules:
RKXMHF: A system that contains RME components, which have the LEGACY_TZ_EN input, will drive a common tie-off input value into all components.
RCLKXF: A PE that supports the LEGACY_TZ_EN tie-off hides the RME capability if LEGACY_TZ_EN is TRUE and reverts all functionality defined by RME.

Scenario:
When Legacy TZ_EN=1 all components including PEs, SMMU hides RME features. System ACS tests will do ID register check.

Scenario ID: 1.19

Rules:
RZHBBL: The memory-mapped registers of a Root watchdog are in the Root PAS.
RVXGBP: A Root watchdog can trigger an RME system reset when predefined expiration conditions are met.

Scenario:
Programming of Root watchdog (if available) from ROOT state will be successful. The test will generate a watchdog interrupt.

Scenario ID: 1.20

Rules:
RZHBBL: The memory-mapped registers of a Root watchdog are in the Root PAS.
RVXGBP: A Root watchdog can trigger an RME system reset when predefined expiration conditions are met.

Scenario:
Programming of Root watchdog (if available) from non-ROOT state (From Non-secure EL2) will be unsuccessful.
Note: Root watchdog memory mapped registers are not accessible from secure PAS.

Scenario ID: 1.21

Rule:
RKGDVK: A Resource can be associated with a PAS using a Granule Protection Table if there is only a single PA within each PAS through which the Resource can be reached and the value of the PA is the same across all physical address spaces.

Scenario:
Check address range of resources are not physically aliased.

Scenario ID: 1.22

Rule:
RKGDVK: A Resource can be associated with a PAS using a Granule Protection Table if the Resource can be assigned to a PAS at page granularity.

Scenario:
Check address range of resources (needs to be protected by GPT) are aligned to page granularity.

Scenario ID: 1.23

Rule:
RHCGZN: If LEGACY_TZ_EN is TRUE, Root PAS is driven to secure PAS by any logic that enforces the PAS Access Table.

Scenario:
When Legacy_TZ_En = True, all Root registers (Interconnect registers SAM registers, DMC- DRAM memory controllers, Timer register) that controls global functionality must be accessible using secure PAS only.

Scenario ID: 1.24

Rule:
RQYRGG: MSD and RMSD are provided with a private interface for accessing a True Random Number Generator (TRNG) that meets the certification profile of the system.

Scenario:
Check that all application PEs support FEAT_RNG or FEAT_RNG_TRAP.

3.2. SMMU

Scenario ID: 2.1

Rule:
RNJRPC: An SMMU in an RME system complies with the Arm® System Memory Management Unit Architecture supplement - The Realm Management Extension (RME), for SMMUv3. (ARM IHI 0094) Arm Ltd.

Scenario:
SMMU must implement RME.

Scenario ID: 2.2

Rule:
RJDBCS: An MMU-attached PAS filter in a non-ACTIVE mode either continues to respond to GPT cache invalidations or invalidates any cached state when moving back to ACTIVE mode.

Scenario:
Change mode of PAS filter to In-Active (if supported). Verify that in In-active mode it responds to GPT cache invalidate.
Algorithm: PWR_Down_SMMU → Invalidate GPT → PWR_UP_SMMU. Observe new GPI value.
Note: This can be evaluated only when SMMU can be powered down.

3.3. PAS Filters

Scenario ID: 3.1

Rule:
RDQTSG: An MPE or a PAS filter in a non-ACTIVE mode in which context is not fully retained blocks its operation and does not service requests until it is in ACTIVE mode again.

Scenario:
Change ACTIVE mode of PAS filter (if supported). Access PA range that is monitored by PAS filter.

Observation:
Read of protected regions does not return data.

Scenario ID: 3.2

Rules:
RBJVZS: An access to a Resource is associated with an Access PAS in accordance with the PAS Access Table.
RYKVJK: A PAS filter enforces the PAS protection check by permitting access to a Resource only if the Access PAS matches a Resource PAS with which that the Resource is associated.
RGDVSZ: A PA of an access to a memory-mapped peripheral is associated with a PAS until reaching the PAS filter assigned to protect the peripheral

Scenario: All protected memory regions are accessible only when resource PAS & access PAS are same.

Note: If Regions are protected by completer side PAS filter, then verify with GPC disabled.

Observation:
Accesses with same resource PAS and access PAS are successful.

Scenario ID: 3.3

Rule:
RGFGZM: If a requester-side Granular PAS filter is in reset state, any requester that is associated with it is either in reset state or blocked from accessing memory.

Scenario:
If SMMU is in reset state it blocks all memory access requests from the devices attached to it.

Observation:
DMA accesses from Exerciser is blocked.

3.4. GIC

Scenario ID: 4.1

Rule:
Scenario:
GIC ITS memory accesses are only to non-secure memory.
Program ITT table base with Root PA and generate access using ITS commands.

Observation:

Expect faults for all the above accesses.

3.5. System reset

Scenario ID: 5.1

Rule:
RCSSDG: MSD SMEM is in the Root PAS.

Scenario:
Access using Root access PAS to Root SMEM is successful after Reset.
Observation:
No fault occurred.

Scenario ID: 5.2

Rule:
RZQQSQ: SMEM that can be dynamically assigned to the Realm PAS, or the Secure PAS is either immediately assigned to the Root PAS or scrubbed on an RME system reset.

Scenario:
Verify Realm SMEM does not reveal old data after system reset.

Observation:
Returned data is not DATA1.

Scenario ID: 5.3

Rule:
RKKSQB: An RME system reset propagates to PEs as either a Cold reset, Warm reset, or Error recovery reset.

Scenario:
RME system reset propagates to all application PEs.
Stimulus: Write non-reset value to SCTLR_EL1/any other system register for all PEs. Apply system reset and check that the system register value is reset.

Scenario ID: 5.4

Rule:
RKQLKN: LEGACY_TZ_EN is not permitted to change value after RME system reset has been deasserted.

Scenario:
LEGACY_TZ_EN = enable does not take effect when system reset is de-asserted.

Scenario ID: 5.5

Rule:
RKQLKN: LEGACY_TZ_EN is not permitted to change value after RME system reset has been deasserted.

Scenario:
LEGACY_TZ_EN = enable take effect after a system reset asserted.

Scenario ID: 5.6

Rule: RNULL

Scenario:
In Realm state all system registers & general purpose registers are scrubbed after reset.

Note:
This scenario is identified in the investigation, so Rule ID, RNULL is used.

3.6. Exerciser

Scenario ID: 6.1

Rule:
RMZJXC: Every requester in the system is subjected to the PAS protection check.

Scenario:
PCiE devices are subject to PAS protection check.

Observation:
DMA transactions to secure, root and realm memory will generate fault.

3.7. Device Assignment

Scenario ID: 7.1

Rule:
RDVJRV: The RME-DA DVSEC is implemented in compliance with PCIe [9] and has the following format:

Scenario: For each function, read the RMEDA registers (DA Capability) and check the corresponding values and its attribute matches the specification.

Observation: No mismatch in both values and attribute properties of the registers

Scenario ID: 7.2

Rule:
RNJRPC: An SMMU in an RME system complies with the SMMU for RME specification [8] and, if the system supports RME-DA or MEC, with SMMU for RME-DA [4].

Scenario: For each SMMU in the system, check if the ROOT_IDR0 register has RME_IMPL set.

Observation: The expected bit values should be set in SMMU

Scenario ID: 7.3

Rule:
RLGXBX: An RME-DA Root Port sets the TEE-IO Supported bit in the Device Capabilities Register.

Scenario: For all Root Ports in the system, the TEE-IO supported bit in the PCIe Extended Capability register should be set.

Observation: The TEE-IO bit should be set

Scenario ID: 7.4

Rule:
RGRCKL: An RME-DA Root Port supports the following IDE features:
• At least one Selective IDE Stream.
– NUM_SEL_STR denotes the number of Selective IDE Streams supported by the Root Port.
• At least three Address Association registers for each Selective IDE Stream.
• The TEE-Limited Stream IDE capability.

Scenario: For all RootPorts in the system, check at least one Selective IDE Stream is supported and TEE-Limited Stream is supported in the IDE Capability register. Check at least three Address Association registers in the Address association block.

Observations: The RootPort should have all the expected values required for the IDE feature.

Scenario ID: 7.5

Rule:
RDVJRV: The RME-DA DVSEC is implemented in compliance with PCIe [9] and has the following format.

Scenario: Check the attribute of the RMEDA_CTRL register.

Observations: The RSVDP fields and RW fields should behave as expected.

Scenario ID: 7.6

Rule:
RMDPKR: When P2P traffic between two TDISP devices is routed through the Root Complex, then for any non-posted request that is forwarded by the Root Complex from a source peer to a target peer, the Root Complex must guarantee that the corresponding completion will be forwarded back to the source peer only if it arrived from the target peer.

Scenario: Get two Exerciser EPs under two different RPs. Transition both the exerciser into TDISP RUN state. Perform a Peer-to-Peer transaction

Observations: Check the competition is obtained only after it is arrived from the target peer.

Scenario ID: 7.7

Rule:
RDVKPF: An outgoing request that has to be sent with IDE-Tbit==1 but that cannot be associated with a Selective IDE Stream that is Locked and in the IDE Secure state, is rejected with error by the RP.

Scenario: For each function, If it is a RP, get the Endpoint BAR Base below it if it is available. Otherwise use the RP's BAR address. Map the BAR to Root PAS and read the data at BAR address from Root world.

Observations: The request should be rejected by the RootPort.

Scenario ID: 7.8

Rule:

RKZBHV: When RMEDA_CTL1.TDISP_EN==1, the RP permits an incoming request to have IDE-Tbit==1 if it arrived on a Selective IDE Stream that is Locked and in the IDE Secure state or if this is enabled by an IMPLEMENTATION DEFINED configuration that is controlled by MSD firmware or a Trusted subsystem, and otherwise rejects the request.

Scenario: Establish an IDE stream in the RP and set the TDISP_EN to 1. Ensure the stream is in secure state. Lock the corresponding Selective IDE register block in RMEDA_CTL2 register. Map the configuration address before writing as REALM PAS. Perform a DMA transaction with IDE-Tbit = 1. Generate a transaction with IDE-Tbit=0 should be rejected by RP.

Observations: The incoming request should be permitted by the RP when IDE-Tbit = 1 and should be rejected when IDE-Tbit = 0.

Scenario ID: 7.9

Rule:
RNPGJV: The RMEDA_CTL registers are RMSD write-protect by hardware default.

Scenario: Read the RMEDA_CTL registers and check if they can be updated from the Root world. Also check if they cannot be updated from the Secure and Non-Secure world.

Observations: RMEDA_CTL registers should behave as write-protect.

Scenario ID: 7.10

Rule:
RPJGJK: When an RP encounters an uncorrectable data integrity error it must do one of the following:

- Poison any TLP that is affected by the error
- Transition any IDE Stream that the affected TLPs are associated with or could be associated with to the IDE Insecure state.

Scenario: Establish an IDE stream between the Exerciser EP and its RP. Inject an error from the exerciser which reaches the RP

Observations: The IDE stream should be transitioned to insecure state from secure state

Scenario ID: 7.11

Rule:
RHCMWC: When RMEDA_CTL1.TDISP_EN transitions from 1 to 0 all hosted IDE Streams transition to IDE Insecure state.

Scenario: After enabling the TDISP_EN, establish the IDE stream between the RP and EP. Once done, set the TDISP_EN to 0

Observations: Check if the IDE stream is transitioned to Insecure state.

Scenario ID: 7.12

Rule:
RYHQQL: When a Selective IDE register block is Unlocked (SEL_STR_LOCK is 0):
- The block registers do not have any register security property
- The associated Selective IDE Stream is in Unlocked state

When a Selective IDE register block is Locked (SEL_STR_LOCK is 1):
- The block registers are RMSD write-detect
- The associated Selective IDE Stream is in Locked state

Scenario: Configure IDE stream between RP and EP and set it to Secure state. Lock the Selective IDE register block by setting SEL_STR_LOCK to 1. Re-Configure the IDE stream

Observations: Check that the IDE stream is transitioned to Insecure state which validates the RMSD write-detect property

Scenario ID: 7.13

Rule:
RRNQNM: When RMEDA_CTL1.TDISP_EN==0:
- The RP rejects an incoming request if it has IDE-Tbit==1 .
- The RP rejects with error an outgoing request if it would otherwise need to be sent with IDE-Tbit==1.

Scenario: Disable the TDISP_EN bit in the RP. Configure the exerciser EP under the RP to TDISP RUN state (IDE-Tbit = 1). Perform a DMA transaction from the Exerciser EP to NS memory.
Map the BAR of the Exerciser EP to ROOT PAS. Perform a read from PE from ROOT.

Observations: Check if both the transaction are rejected and should be unsuccessful

Scenario ID: 7.14

Rule:
RMJNLW: Requests that are autonomously initiated by the RP over its host interface are tagged with PAS==Non-secure. Likewise, a request initiated by the RP over the PCIe interface must have IDE-Tbit==0.

Scenario: Map the GIC ITS ITT base to ROOT PAS. Generate an MSI from the RP by injecting an error in RP.

Map the GIC ITS ITT base to NON-SECURE PAS. Generate an MSI from the RP by injecting an error in RP.

Observations: Check that the interrupt is not serviced in NS world when ITT is mapped to ROOT. Check that the interrupt is serviced in NS world when ITT is mapped to NS.

Scenario ID: 7.15

Rule:
RPCRFM: When RMEDA_CTL1.TDISP_EN==1 the following registers are RMSD write-detect:

- RP configurations that are not allowed to be modified when the RP has an IDE Stream bound to a TDI as specified in TDISP [8].
- IMPLEMENTATION DEFINED registers that can impact the RME security guarantee and that must be programmed by Non-secure state.
    - For example, RP registers that perform address translation between system hardware address space and PCIe address space.

Scenario: Establish an IDE stream between RP and EP. The IDE stream should be in secure state. Modify the RP configuration registers

Observations: Check the write-detect property by ensuring the IDE stream is transitioned to Insecure state

Scenario ID: 7.16

Rule:
RMYKFH: When an RP forwards an incoming request over a host interface it sets the SMMU SEC_SID, StreamID and SubstreamID fields as follows:

- If the request has IDE-Tbit==1, SEC_SID is set to 0b10 (Realm). Otherwise SEC_SID is set to 0b00 (Non-secure).
- SMMU StreamID and SubstreamID are set using the RID and PASID fields in accordance with Arm BSA [15] and SBSA [16] specifications.

Scenario: Generate a transaction with IDE-Tbit = 1 and map the destination address as Realm PAS in SMMU. Generate a transaction with IDE-Tbit = 0 and map the destination address as Non-Secure PAS in SMMU.

Observations: The transaction should be successful. Covered as part of test 608.

Scenario ID: 7.17

Rule:

RGKHSZ: An RME-DA RP performs the following operations for all outgoing TLPs:

- Associate the TLP with an IDE Stream.
- Set the IDE-Tbit of the TLP to the appropriate value.

Scenario: Generate a DMA transaction on the correct IDE stream and with appropriate IDE-T bit (0 for NS and 1 for Realm).

Observations: The transaction should be successful if it arrives on the correct IDE stream and the successful transaction confirms the IDE-Tbit values. This scenario is covered by the tests 608 and 613 generating DMA transactions with IDE-Tbit as 0 and 1.

Scenario ID: 7.18

Rule:
RDNFTD: A PA of an access to a PCIe Root Port is associated with a PAS until reaching the Root Port.

Scenario: This rule would be implicitly checked by RME-DA rules for T-bit value (seeing that PAS=Realm converts to T=1)

Scenario ID: 7.19

Rule:
RNWSJB: All Root Ports in an RME-DA system must implement the RME-DA DVSEC

Scenario: This rule would be implicitly checked by RME-DA rule RDVJRV as part of test 601 and 605 where we check for the PCIe DVSEC registers

Scenario ID: 7.20

Rule:
RGSTJC: Any of the following events transitions all hosted IDE Streams to IDE Insecure state:

- A reset or loss of state of a write-detect, write-protect or full-protect register.
- A reset or loss of state of a Root Port component that affects the RME security guarantee.

Scenario: This rule would be implicitly checked by RME-DA rule RPCRFM where Transitions related to RME Security guarantee.

Scenario ID: 7.21

Rule:
RZJJMZ: As a requester, an RCiEP sets the SMMU SEC_SID, StreamID and SubstreamID fields of a request as follows:

- If the request must be sent with IDE-Tbit==1, the RCiEP sets SEC_SID to 0b10 (Realm). Otherwise the RCiEP sets SEC_SID to 0b00 (Non-secure).
- SMMU StreamID and SubstreamID are set using the RID and PASID fields in accordance with Arm BSA [15] and SBSA [16] specifications.

Scenario: This rule is covered as part of test 608 and 615

Scenario ID: 7.22

Rule:
RCFQBW: IDE-Tbit for an outgoing PCIe Memory Request or Configuration Request is set based on the request PAS: If PAS is Realm or Root then IDE-Tbit is 1 and otherwise it is 0.

Scenario: Retrieve the BAR of the Endpoint (skipping this step if the Endpoint lacks an MMIO BAR), identify the RootPort for the Endpoint, and enable the TDISP_EN bit in the RME-DA DVSEC register. Map the BAR address to Realm PAS, establish an IDE Stream between the RootPort and Endpoint, and transition the Endpoint to the TDISP RUN state. Perform write and read operations at the BAR address from the Realm world.
Additionally, retrieve the BAR of the Endpoint (skipping if it lacks an MMIO BAR), enable the TDISP_EN bit in the RME-DA DVSEC register, and map the BAR address to Non-Secure PAS. Perform write and read operations at the BAR address from the Root world.

Observations: The request should be accepted by the RootPort, confirming that the IDE-Tbit is set appropriately based on the PAS mapping.
The request should be accepted by the RootPort, confirming that the IDE-Tbit is set appropriately based on the PAS mapping. Similarly, the request should also be allowed by the RootPort when the BAR address is mapped to Non-Secure PAS.

Scenario ID: 7.22

Rule:
RGBVTS: As a completer of memory requests a TDISP-compliant RCiEP extracts the request IDE-Tbit from the request PAS: If PAS is Realm or Root then IDE-Tbit is 1, otherwise it is 0.
.

Scenario: Retrieve the BAR of the RCiEP (skipping this step if the RCiEP lacks an MMIO BAR) and enable the TDISP_EN bit in the RME-DA DVSEC register. Map the BAR address to Realm PAS and transition the RCiEP to the TDISP RUN state. Perform write and read operations at the BAR address from the Realm world.
Additionally, retrieve the BAR of the RCiEP (skipping if it lacks an MMIO BAR), enable the TDISP_EN bit in the RME-DA DVSEC register, and map the BAR address to Non-Secure PAS. Perform write and read operations at the BAR address from the Root world.

Observations: The request should be accepted by the RCiEP, confirming that the IDE-Tbit is set appropriately based on the PAS mapping.
The request should be accepted by the RCiEP, confirming that the IDE-Tbit is set appropriately based on the PAS mapping. Similarly, the request should also be allowed by the RCiEP when the BAR address is mapped to Non-Secure PAS.

Scenario ID: 7.23

Rule:
RXHMDQ: When RMEDA_CTL1.TDISP_EN==1 the following registers are RMSD write-protect:

- IMPLEMENTATION DEFINED registers that can impact the RME security guarantee and that are programmed by MSD firmware or a Trusted subsystem. For Example:
    - Registers that allow reading or modifying any Transaction Layer Packet (TLP) parameters, such as its address or data, or that may lead to a drop, corrupt, replay or reorder of a TLP,
        - Before IDE is applied (for outgoing TLPs) or,
        - After the IDE check (for incoming TLPs).
    - Registers that allow forwarding a Poisoned TLP as a non-Poisoned TLP.
    - Registers that define the method of signaling an Unsupported Request (UR) over the host interface.
    - A register that controls the Root Port ID or the PCIe Segment Number of the Root Port.
    - Registers that may affect the correctness of IDE functionality, for example error injection controls.

Scenario: Verify that the implementation-defined root port registers identified as RMSD write-protect are only writable when the RMEDA_CTL1.TDISP_EN register is disabled. When TDISP_EN is enabled, validate that these registers are protected against write access from NS.

Observations: When RMEDA_CTL1.TDISP_EN is enabled, any attempt to write to these RMSD write-protect registers from NS must fail with an appropriate fault or error.

Scenario ID: 7.24

Rule:
RNXJKQ: When RMEDA_CTL1.TDISP_EN==1 the following registers are RMSD full-protect:

- IDE key programming registers.
- Registers that store IDE confidential information, for example Initialization Vectors (IV) or IMPLEMENTATION DEFINED confidential state.
- Registers that store payload from TLPs that have IDE-Tbit==1.

Scenario: Validate that the root port registers identified as RMSD full-protect are inaccessible from NS. The addresses of these registers are retrieved from the PAL, and their full-protect behavior is tested by attempting both read and write operations.

Observations: Any attempt to read from or write to these RMSD full-protect registers must fail with an appropriate fault or error, ensuring that they remain fully protected.

Scenario ID: 7.25

Rule:
RTTPLM: Interconnect registers that control mapping of PAs to PCIe Root Ports are implemented as MSD-Protected registers.

Scenario: Validate that the interconnect registers responsible for mapping PAs to PCIe Root Ports are implemented as MSD-Protected registers and ensure that they are accessible exclusively from the MSD domain. Retrieve the register addresses as provided by the PAL implementation and attempt to access them from both MSD and non-MSD domains.

Observations: Access to the registers should succeed when performed from the MSD domain, whereas access from non-MSD domains should fail with an appropriate fault or error.

3.8. Device Permission Table

Scenario ID: 8.1

Rule:
RQRMPD: A translated access from a TDI that is assigned to Realm state is subject to DPT checks, unless where stated otherwise.

Scenario: Validate that IDE-tagged transactions from the Exerciser Endpoint undergo proper DPT enforcement through the R_SMMU. Establish an IDE stream between the Root Port and Exerciser, configure secure EL3 memory for DMA, and evaluate both successful and failed flows based on whether a DPT Invalidate command is issued. The test ensures that transactions with stale or missing DPT entries are blocked and those with valid, updated entries are allowed.

Observations: The transaction initiated by the Exerciser passed through the R_SMMU and was subjected to DPT checks as expected.

Scenario ID: 8.2

Rule:
RPGSTQ: An RME system can include on-chip TDISP-compliant devices that are measured and attested by HES or MSD. For such a device:

- DPT checks can be skipped. • GPC cannot be skipped.

Scenario: This scenario is identical to Scenario ID 8.1, with the only difference being that the requestor device is an RCiEP (Root Complex integrated Endpoint) instead of a standard Endpoint.

Observations: The transaction initiated by the RCiEP Exerciser passed through the R_SMMU and was subjected to DPT checks as expected.

3.9. Memory Encryption Contexts

Scenario ID: 9.1

Rule:
R_{BJVZS}: An access to a Resource is associated with:
- A MECID, in accordance with the rules specified in [1] and [3].

I_{XQKRQ}: Arm Recommends that all RME system components support the same MECID width, to avoid faulty behavior.

Scenario:
- Check that all requesters (PEs and SMMUs) support MEC
- Read MECID width of all the requesters and establish a common MECID width - MECIDW.
- Check that both 2^(MECIDW - 1) and 2^(MECIDW - 2) works
  - Map VA to PA in Realm PAS.
  - Enable MEC.
  - Write data to VA with MECID as 2 ^ (MECIDW - 1) and issue CMO to PoPA/PoE.
  - Read VA and store as data1
  - Write data to VA with MECID as 2 ^ (MECIDW - 2) and issue CMO to PoPA/PoE.
  - Read VA and store as data2.

Observation: All requesters support MEC and data1 != data2

Scenario ID: 9.2

Rule:
R_{TBZM}: An access to a cacheable memory Location is associated with a MECID until reaching the PoE.
R_{MLFBL}: External memory assigned to Secure PAS, Realm PAS, or Root PAS must be encrypted using a method that provides at least all the following:
- In a system with MEC, a different encryption context for each MECID in the Realm PAS.

R_{MYWVB}: Data is encrypted before being written to external memory or to any shared cache that resides past the PoPA. In a system with MEC, data is encrypted before being written to external memory or to any shared cache that resides past the PoE

Scenario:

- MAP VA to PA in Realm PAS.
- Write to VA with data1 with MECID1
- Issue CMO to PoPA/PoE and Read VA with MECID2 store in data2
- Perform similar DMA transaction from a PCIE device to validate SMMU MECID tagging.

Observation: data1 != data2

Scenario ID: 9.3

Rule:
R_{QBNJF}: A PoPA CMO affects any cached copy in the system with the specified {PAS, PA} regardless of all of the following:

- The MECID that it was cached with, in a system with MEC

I_{MNGJT}: In an RME system with MEC, RLCXDB also applies to any cached or transient state associated with the PA before the PoE.

Scenario:

- Map VA to PA in Realm PAS.
- Enable MEC, Sect MECID = MECID1
- Write data1 to VA
- Change MECID = MECID2, Issue CMO(clean and invalidate) to PoPA
- Mark VA as non-cacheable.
- Change MECID back to MECID1
- Read VA == data1 (indicates cache was cleaned and regardless of MECID being MECID2 while issuing CMO)
- Write data2 to VA.
- Mark memory as cacheable.
- Read VA == data2 (indicates cache was invalidated)

Observation: Reads to VA in the above steps are as specified in the scenario.

Scenario ID: 9.4

Rule:
R_{KMNQX}: Memory accesses resulting from a cache clean operation, due to cache maintenance operations and natural evictions, use the MECID that the entry was cached with.

Scenario:

Multi PE Variant 1:

Primary PE	Secondary PE
• Enable MEC, Set MECID1 • Map VA to PA in Realm PAS • Write data1, Issue CMO to PoC • Set MECID2, issue CMO to PoC	• Map VA to PA in Realm PAS • Enable MEC, Set MECID1 • Read VA, read data == data1

Multi PE Variant 2:

Primary PE	Secondary PE
• Enable MEC, Set MECID1 • Map VA to PA in Realm PAS • Write data1	• Map VA to PA in Realm PAS • Enable MEC, Set MECID2 • Issue CMO to PoC • Set MECID1 • Read VA, read data == data1

Single PE scenario:

- Enable MEC, Set MECID1
- Map VA to PA in Realm PAS
- Write data1, Issue CMO to PoC
- Set MECID2
- Issue CMO to PoC
- Set MECID1
- read VA, read data == data1
- Repeat the above for CMOs to PoE and PoPA.

Observation: Reads to VA in above steps are as specified in the scenarios.

4. Out of scope rules

4.1. System PMU counters

Rule:
RHRVJB: A system PMU counter that is accessible in the Secure PAS can only count events that are attributable to the Secure PAS or to the Non-secure PAS.
Rule:
RBSZPN: A system PMU counter that is accessible in the Realm PAS can only count events that are attributable to the Realm PAS or to the Non-secure PAS.
Rule:
RTMSNN: A system PMU counter that is accessible in the Root PAS can count events that are attributable to any PAS.
Rule:
RMMPWY: A system PMU counter that is accessible in the Non-secure PAS can count events that are attributable to a specific PAS if there is a per-PAS authentication control that can permit events from that PAS to be counted.
Rule:
RPLXZB: A per-PAS authentication control can be driven by a debug authentication interface signal or by a register accessible in the corresponding PAS or in the Root PAS.
Rule:
RCFYKS: An event that is not explicitly associated with a PAS but can leak confidential information is implicitly associated with the Root PAS.

4.2. Debug

Rule:
RQSXBZ: RMSD external debugging and Root external debugging are disabled by default on a Secured Arm CCA system.
Rule:
RHLTLK: RMSD external debugging can only be authorized following an RME system reset and before RMSD firmware is loaded and cannot change state until a subsequent RME system reset.
Rule:
RXVNFV: Root external debugging can only be authorized following an RME system reset and before MSD firmware is loaded and cannot change state until a subsequent RME system reset.
Rule:
RGTPGZ: When Root external debugging is enabled, the RNVS confidential parameters are either inaccessible, scrubbed, or populated with debug values.
Rule:
RRHGKX: Access to a Secured Arm CCA system through an external debug or test interface, including debug access ports, JTAG ports, and scan interfaces is disabled by default. Debug access

can be enabled following validation of a debug certificate or password which is injected via an external debug interface.

Rule:
RQLPNL: When external debugging is enabled for any Security state, external requests to power-up a component within a level of the system hierarchy (PE, PE-Cluster, System) are permitted but must be executed by trusted power control.

4.3. Hardware enabled security

Rule:
RNWQBJ: If HES is hosted as a tenant within a multi-tenant Trusted subsystem, HES functionality must be isolated from other tenants, such that tenants must not be able to monitor HES functionality or impact HES functionality or integrity.

Rule:
RHJSSG: The HES implementation exposes a private interface to SSD components such as Trusted subsystems for requesting HES services.

Rule:
RCGDVX: The HES implementation exposes a programming interface in the Root PAS, shared by all application PEs, allowing MSD and PE Initial boot ROM to request for HES services.

Rule:
RBQPFG: HES has exclusive read and write access to RNVS confidential parameters.

Rule:
RBTWVY: A measurement register can be either extended using a secure hash algorithm, locked, or reset.

Rule:
RDFPJL: HES has exclusive access to extend, lock, and reliably obtain the value of a measurement register it owns.

Rule:
RFWSRF: Once locked, a measurement cannot be further extended until it is reset.

Rule:
RWYSLK: An RME system reset is the only method to reset a measurement owned by HES.

Rule:
RXCRMH: On an RME system reset, HES state is reset to a known value, including all measurements and ephemeral cryptographic context.

4.4. RAS

Rule:
RGNGMB: Only SSD or MSD can control whether recording is performed for error records that might contain confidential information.

Rule:
RGZTVL: Critical Error Interrupts (CI) must be wired to a Trusted subsystem that will respond with an RME system reset.

Rule:
RLWVCX: An uncontainable error results in an RME system reset.
Rule:
RJNBWJ: Only SSD or MSD can enable or disable the generation of a CI.
Rule:
RXPCTR: Where an MPE provides support for integrity, if it detects an integrity error it can perform one of the following responses:

- Respond by returning poison back to the consumer and record the error as a deferred error.
- Respond with an in-band error response and record the error as an uncorrected error.

Rule:
RHSVLQ: Only SSD or MSD must be able to control the abilities of detecting, propagating, and reporting MPE integrity errors.
Rule:
RGZHTD: In addition to providing encryption and, where implemented, integrity capabilities, the MPE can pass poison information:

Note: If a requester above the MPE defers errors by writing poison, then the MPE must be able to pass this value through to the memory system below it as poison.
If a requester above the MPE consumes a memory location that has been marked as poison, either because of that access or a previous access, the MPE must pass that poison to consumer.

4.5. RNVS

Rule:
RWNPYD: A programming interface that allows read and write access to RNVS must be in the Root PAS.
Rule:
RLMSSL: The system supports a method for permanently blocking read access from application PEs to RNVS confidential parameters.
Rule:
RVXBYG: System support for any memory protection property reported in System Properties is immutable and applicable for all DRAM memory controllers in the system.

4.6. Trusted System Control Processor

Rule:
RSXCFK: A Trusted SCP is an on-chip control processor that is trusted by MSD and can access resources in the Root PAS.
Rule:
RZHJQJ: A Trusted SCP is considered a Trusted subsystem and must meet the applicable security requirements, for example, supporting Secure boot and having attestable firmware.
Rule:
RMZDXV: It is permitted for a Trusted SCP to have a mechanism to bypass a PAS filter which filters

its transactions.

4.7. Miscellaneous

Rule:
RDFYXL: In an RME system, any access by a requester and any instruction executed by a PE is associated with a single Security state.
Rule:
RQDWVC: Either SSD or MSD controls Association of a Resource with a Resource PAS.
Rule:
RSCDLL: Once assigned, the value of an Access PAS cannot be altered.
Rule:
RWRGTF: Access to the Root PAS is only permitted for Trusted requesters.
Rule:
RWJNMD: Granule Protection Check for on-chip Resources can only rely on Granule Protection Tables that are stored on-chip or are stored off-chip with equivalent level of integrity and replay protection.
Rule:
RGQCQT: A Granule Protection Check that applies to non-idempotent locations does not permit any access to be speculatively performed to a non-idempotent location before the Granule Protection Check for the access is complete.
Specification Rule:
RMYWVB: Data is encrypted before being written to external memory or to any shared cache that resides past the PoPA.
Rule:
RBNSQB: An ECC-scrubbing engine located after the PoPA must not leak confidential information, for example through error record registers.
Rule:
RRHBJN: The Security state of a non-PE requester that is not a Trusted subsystem can be either Secure or Non-Secure state.
Rule:
RMCMSH: A fully coherent non-PE requester, which is not part of the System Security Domain (SSD), will not observe coherent traffic for addresses in the Secure, Realm, or Root PAS.
Rule:
RRGQRT: If a programmable completer-side PAS filter can assign resources to all physical address spaces then:

- The registers that control the filter are in the Root PAS.
- On an RME system reset, Resources controlled by the filter are either assigned to the Root PAS or are reset to a known value.

Rule:
RGLLZY: If a programmable completer-side PAS filter assigns resources only to the Secure PAS and Non-secure PAS then:

- The registers that control the filter are in the Secure PAS or in the Root PAS.

- On an RME system reset, Resources controlled by the filter are either assigned to the Secure PAS or the Root PAS or are reset to a known value.

Rule:
RJSDVG: All RME structures and fields use little-endian convention.
Rule:
RSPLKT: The address ranges of MSD SMEM are either defined statically or defined by SSD following an RME system reset.
Rule:
RZVQGS: The address ranges of SMEM assigned to the Realm PAS and Secure PAS are either defined statically or by SSD or MSD.
Rule:
RZCJHY: The access control path that protects SMEM is not affected by state from non-shielded memory.
Rule:
RXBKYB: All bus and interconnect decoding components between the point where the Access PAS is assigned and the PoPA are PAS tag aware.
Rule:
RLCXDB: Completion of a PoPA CMO for a given PA guarantees that both:

- Any dirty cached or transient state associated with the PA before the PoPA has been cleaned to after the PoPA.
- Any cached or transient state associated with the PA before the PoPA has been invalidated.

Rule:
RCMMDG: For any cache before the PoPA, cache prefetching across granule-boundary is allowed only after querying the GPC for the PAS association of the next granule.
Rule:
RPSGCM: A cache maintenance operation performed on a clean cache entry never results with a write of entry content past the PoPA.
Rule:
RKSPKN: Encryption keys used by MPE are stored in registers that are reset to a known default value on an RME system reset.
Rule:
RYHXPH: An MPE integrity error is reported as an external abort to a software or hardware agent consuming the error.
Rule:
RYJDSJ: Any captured details of an MPE integrity error are only visible to MSD.
Rule:
RLPQSN: An MPE property that is reported through the System Properties structure in Root Non-volatile Storage (RNVS) is supported for all external memory ports in the system.
Rule:
RVDFYZ: A register that is located outside of the Root PAS but can affect a service provided by MSD must be implemented as a measurable register.
Rule:
RYLVDB: A measurable register is a write-lockable register that MSD has a trusted method to obtain its value.
Rule:
RRFSYB: An RME system propagates a 2-bit MPAM_SP field to all MSCs that are either a Four-space

MSC or have a PARTID space mapper.

Rule:

RCFYBJ: An IMPLEMENTATION DEFINED property of an architecture extension, or an IMPLEMENTATION DEFINED difference between application PEs must not create an exposure that could break the RME security guarantee.

Rule:

RXKBNZ: PE behavior is UNPREDICTABLE when the following are true:
An IMPLEMENTATION DEFINED difference between application PEs is visible to software, for example through different System register values across PEs.
There is a mismatch between the register value assumed by software running on a PE and the actual hardware value of the PE.
An example where such mismatch could occur, is if software obtained the value by reading it on a different PE.

Rule:

RLRQXZ: A software-initiated power state transition in an RME system at any level of the system hierarchy (PE, PE-cluster, System) is validated by MSD or by a Trusted subsystem.

Rule:

RWJVRX: Save/Restore operations for MSD PE context can only be done by MSD or a Trusted subsystem and use storage that is not accessible from Realm, Secure and Non-secure states.

Rule:

RMVZHF: Save/Restore operations for RMSD PE context can only be done by RMSD, MSD, or a Trusted subsystem and use storage that is not accessible from Secure and Non-secure states.

Rule:

RRCLYM: Save/Restore operations for PE context of Secure state can only be done by MSD or a Trusted subsystem or software running in the Secure state and use storage that is not accessible from Realm and Non-secure states.

Rule:

RGVJYZ: Any register that affects a system power policy or a hardware power mode is implemented as an MSD-Protected Register (MPR).

Rule:

RKYXMR: Any power management operation that can affect MSD state or the RME security guarantee must be validated by MSD or a Trusted subsystem.

Rule:

RHJHRL: On an RME system reset, all Trusted requesters and Trusted subsystems are reset. Any Trusted subsystem state that might include MSD or RMSD confidential information is reset to known values.

Rule:

RHLKZP: An RME system reset might propagate to any component that implements RAS [6] as an Error recovery reset.

Rule:

RSSGMJ: The reset of a system component that affects the RME security guarantee can only be controlled by MSD or a Trusted subsystem or driven by an RME system reset.

Rule:

RCKBGZ: A legacy completer is attached to an RME IP by driving the NS signal of the completer from PAS [0] of the RME IP.

Rule:

RYKSSD: A legacy requester is attached to an RME IP by driving PAS [0] of the RME IP from the NS signal of the legacy requester and driving PAS [1] of the RME IP to 0b0.

Rule:
RYXFMV: A requester that is accessing memory-mapped resources not through a stage 1 or stage 2 MMU/SMMU must support a method that is enforced by SSD hardware for tagging accesses with an Access PAS, in accordance with the PAS Access Table (Table B2.1).
For example:

- A Debug Access Port (DAP) can expose a programming register to an external debugger that allows setting an Access PAS to one of the permitted values, as implied by the debug authentication interface state, for any access that targets main memory or an APB peripheral.
- If the debug authentication interface permits RMSD external debugging but not Secure external debugging then DAP hardware would reject an attempt to program the register to Access PAS == Secure.
    - Furthermore, if the debug authentication interface permits RMSD external debugging then DAP hardware can permit accesses with Access PAS == Realm to specify a programmed MECID.

Rule:
RLYXGC: A CTC interface in a multi-chip RME system supports all of:

- Transport of the PAS tag with any access that specifies a physical address (PA).
- Transport of the MECID with any access that specifies a PA, if the RME system supports MEC.
- Transport of CMO and DVM messages that RME and MEC [1] specify.

4.8. DA

Rule:
RWBJJT: TSM functionality in RME-DA is implemented within RMSD.
Rule:
RBDLXG: An RME-DA Root Port exposes an IDE key programming interface for the following IDE key management (IDE_KM) data objects:

- KEY_PROG
- K_SET_GO
- K_SET_STOP

Rule:
RVCRRM: An RME-DA Root Port must support IDE key refresh operations in compliance with [13].
Rule:
RFSFST: The RP IDE logic must be able to detect that an IDE key set requires a refresh and perform one or more of the following:

- Assert a dedicated interrupt that will be delivered to a Trusted subsystem.
- Transition the corresponding IDE Stream to Insecure state.

Rule:
RBWFTS: RMSD ensures that Selective IDE Streams are configured such that different IDE

Streams are assigned with RID ranges and address ranges that are not overlapping.

Rule:

RSWBSV: IDE-Tbit of PCIe messages is set as follows:

- For messages generated from DTI requests, IDE-Tbit is extracted from the DTI request in compliance with AMBA DTI Revision 3 (See: SMMU for RME-DA [6]).
- For Vendor-Defined messages, the IDE-Tbit is permitted to be 1 if the RP has a method to associate the message with the Root or Realm Security states.
- For any other message, IDE-Tbit is set to 0. For example, Power Management messages.

Rule:

RCKJMN: IDE-Tbit for PCIe completions is set in compliance with IDE [13] and TDISP [8]. This means that:

- For ATS Translation Requests, the host will set the IDE-Tbit on the corresponding ATS Translation Completion to match the IDE-Tbit value of the request.
- For ATS-translated read requests the host will set the IDE-Tbit value on the corresponding read completion to match the value of the request, with the following exception:
    - If a P2P read request with IDE-Tbit==1 is forwarded through the host to a non-TDISP device, the host is permitted but not required to set IDE-Tbit==0 on the corresponding completion.

Rule:

RLMFSV: When RMEDA_CTL1.TDISP_EN==1, any RP debug functionality that might affect the RME security guarantee is disabled unless explicitly enabled by one of the following:

- An access to a write-protect register.
- An assertion of a debug authentication signal indicating that either RMSD external debugging or Root external debugging are enabled.

Rule:

RQNTYC: The PCIe segment and RIDs that are allocated to an RCiEP are either defined statically or configured using an RMSD write-protect register.