arm

Arm Server Base System Architecture

Revision: r3p1

Arm Base System Architecture Test Scenarios

Non-Confidential

Issue 5

PJDOC-2042731200-3439

Arm SBSA Compliance Suite

Release information

Document history

Issue	Date	Confidentiality	Change
02	05 May 2018	Non-Confidential	Changes from REL 1.0.
03	20 March 2020	Non-Confidential	Changes from REL 2.3 and REL 2.4
04	30 September 2020	Non-Confidential	Changes for REL 3.0
05	27 September 2021	Non-Confidential	Changes for REL 3.1

Non-Confidential Proprietary Notice

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 a product under development.

Progressive terminology commitment

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

This document includes terms that can be offensive. We will replace these terms in a future issue of this document. If you find offensive terms in this document, please email **terms@arm.com**.

Web Address

www.arm.com.

Contents

1 Introduction

1.1 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, r*1*.

p*n*
: Identifies the minor revision or modification status of the product, for example, p*2*.

1.2 Intended audience

This document is for engineers who are verifying an implementation of Arm® Base System Architecture 1.0.

1.3 Conventions

The following subsections describe conventions used in Arm documents.

1.3.1 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: **https://developer.arm.com/glossary**.

1.3.2 Typographical Conventions

Convention	Use
italic	Introduces citations.
bold	Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.
`monospace`	Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.
`monospace` **`bold`**	Denotes language keywords when used outside example code.
`monospace` `underline`	Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
<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, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.

1.4 Additional Reading

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

Table 1-1 Arm publications

Document name	Document ID	Licensee only
Arm® Server Base System Architecture (Version 6.0)	DEN 0029C	No
Arm® Architecture Reference Manual ARMv8, for Armv8-A architecture	DDI 0487F.a	No

Table 1-2 Other publications

Document ID	Document name
None	*PCI Express Base Specification Revision 5.0, Version 1.0*

1.5 Feedback

Arm welcomes feedback on this product and its documentation.

1.5.1 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.

1.5.2 Feedback on content

If you have comments on content, send an email to **support-systemready-acs@arm.com** and give:

- The title Arm Base System Architecture Test Scenario.
- The number PJDOC-2042731200-3439.
- 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.

2 About this document

This document describes the test scenarios for SBSA Architecture Compliance.

2.1 Terms and abbreviations

This document uses the following terms and abbreviations.

Table 2-1: Terms and abbreviations

Term	Abbreviations
ACS	Architecture Compliance Suite
ACPI	Advanced Configuration and Power Interface
LPI	Low Power Interrupt
MSI	Message Signaled Interrupts
PAL	Platform Abstraction Layer
PASID	Process Address Space ID
PE	Processing Element
PMU	Performance Monitoring Unit
PIPT	Physically Indexed Physically Tagged
PPI	Private Peripheral Interrupt
SBSA	Server Base System Architecture
SGI	Software-Generated Input
SMC	Secure Monitor Call
SMMU	System Memory Management Unit
SPI	Shared Peripheral Interrupt
VIPT	Virtually Indexed Physically Tagged

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 configuration system features described in SBSA architecture.

3 Introduction to SBSA

The SBSA specifies a hardware system architecture that is based on Arm 64-bit architecture. The server system software such as operating systems, hypervisors, and firmware can rely on this architecture. It addresses PE features and key aspects of system architecture.

The primary goal is to ensure enough standard system architecture to enable a suitably built single OS image to run on all the hardware compliant with this specification. A driver-based model for advanced platform capabilities beyond basic system configuration and boot are required. However, that is outside the scope of this document. Fully discoverable and describable peripherals aid the implementation of such a driver model.

SBSA also specifies features that firmware can rely on, allowing for some commonality in firmware implementation across platforms.

4 Cross reference to architecture and tests

The tests are divided into a hierarchy of subcategories depending on the runtime environment and the component submodules that are required for achieving the verification. The top level of the hierarchy is consistent with the target hardware subsystem which is validated by the test.

These are compliance level 0 to compliance level 5 as per SBSA specification version 6.0.

A test may check for different parameters of the hardware subsystem based on the level of compliance requested. Also, the tests are further subclassified as required to run in an EL3 environment. The communication between the ACS and the EL3 firmware is through Arm SMC.

The tests are classified as:

- PE
- GIC
- Timer
- Watchdog
- PCIe
- Exerciser
- Wakeup semantics
- Peripherals
- IO Virtualization (SMMU)

4.1 PE

PE tests require the following tests in the table to run all the PEs in the system, requiring a Software-Generated Interrupt (SGI) is broadcast with the test address as an entry point.

Test ID	Test case	System interface dependency	Requirement of Secure firmware	Level
1	Number of PEs does not exceed 2^28.	ACPI MADT table	No	Level 2+
2	PEs implement Advanced SIMD extensions.	CPU System Register Read	No	Level 0+
3	PE implements 16-bit ASID support.	CPU System Register Read	No	Level 0+
4	PE supports 4KB and 64KB at stage 1 and 2.	CPU System Register Read	No	Level 0+
5	Cache is implemented as VIPT or PIPT.	CPU System Register Read	No	Level 0+
6	All PEs are coherent and in the same Inner Shareable domain.	CPU System Register Read	No	Level 0+

Test ID	Test case	System interface dependency	Requirement of Secure firmware	Level
7	PEs must implement cryptography extensions.	CPU System Register Read	No	Level 0+
8	PEs implement little-endian support.	CPU System Register Read and functional	No	Level 0+
10	PEs implement AArch64 at all Els.	CPU System Register Read	No	Level 0+
11	PMU overflow signal from each PE must be wired to a unique PPI or SPI interrupt.	ACPI MADT and functional	No	Level 0+
12	Each PE implements a minimum of six programmable PMU counters.	CPU System Register Read	No	Level 1+
13	Each PE implements a minimum of four synchronous watchpoints.	CPU System Register Read	No	Level 0+
14	Each PE implements a minimum of six breakpoints.	CPU System Register Read	No	Level 1+
15	All PEs are architecturally symmetric except for permitted differences.	CPU System Register Read	No	Level 0+
17	Each PE implements CRC32 instructions.	CPU System Register Read	No	Level 3+
18	If PEs implement SVE and the Statistical Profiling Extension (SPE), it also implements Armv8.5-SPE.	CPU System Register Read and functional	No	Level 6+
19	All PEs must implement the RAS extension introduced in Armv8.2.	CPU System Register Read and functional	No	Level 4+
20	All PEs must implement support for 16-bit VMD.	CPU System Register Read and functional	No	Level 4+
21	All PEs must implement virtual host extensions.	CPU System Register Read and functional	No	Level 4+
22	If PEs implement Armv8.3 pointer signing, the PEs must provide the standard algorithm defined by the Arm architecture.	CPU System Register Read and functional	No	Level 4+
23	All PEs must implement enhanced nested virtualization.	CPU System Register Read and functional	No	Level 5+
24	All PEs must support changing of page table-mapping size using level 1 and level 2 solution proposed in the Armv8.4 extension. Level 2 is recommended.	CPU System Register Read and functional	No	Level 5+
25	All PEs must provide support for stage 2 control of memory types and cacheability, as introduced by Armv8.4 extensions.	CPU System Register Read and functional	No	Level 5+
26	All PEs must implement the Activity Monitors Extension.	CPU System Register Read and functional	No	Level 5+
27	Where export control allows, all PEs must implement cryptography support for SHA3 and SHA512.	CPU System Register Read and functional	No	Level 5+

Test ID	Test case	System interface dependency	Requirement of Secure firmware	Level
28	Hardware updates to Access flag and Dirty state in translation tables must be supported.	CPU System Register Read and functional	No	Level 6+
29	PEs must implement restrictions on speculations introduced in the Armv8.5 extensions	CPU System Register Read and functional	No	Level 6+
30	PEs must implement Speculative Store Bypass Safe	CPU System Register Read and functional	No	Level 6+
31	PEs must implement the SB speculation barrier.	CPU System Register Read and functional	No	Level 6+
32	PEs must implement the CFP RCTX, DVP RCTX, and CPP RCTX instructions.	CPU System Register Read and functional	No	Level 6+
33	PEs must provide support for Branch Target Identification.	CPU System Register Read and functional	No	Level 6+
34	PEs must protect against timing faults that are used to guess page table mappings.	CPU System Register Read and functional	No	Level 6+
35	PEs support enhanced virtualization traps.	CPU System Register Read and functional	No	Level 6+
36	All PEs implement Armv8.5-PMU.	CPU System Register Read and functional	No	Level 6+

4.2 GIC

GIC functionality is verified from running the test on a single PE in the system.

Test ID	Test case	System interface dependency	Requirement of Secure firmware	Level
101	GICv2 is implemented.	ACPI, register read	No	Level 0,1
-	GICv3 is implemented.	-	-	Level 2+
102	If the base server system includes PCIe, then the GICv3 interrupt controller implements ITS and LPI.	MADT Table	No	Level 2+
103	The GICv3 interrupt controller supports two Security states.	GIC System Register Read	No	Level 3+
104	GIC maintenance interrupt is wired as PPI 25.	ACPI Table	No	Level 2+

4.3 Timer

Timer functionality is verified from running the test on a single PE in the system.

Test ID	Test case	System interface dependency	Requirement of Secure firmware	Level
201	The system counter of the Generic Timer runs at a minimum frequency of 10MHz and at a maximum frequency of 400MHz.	ACPI GTDT	No	Level 0+
202	The local PE timer must generate a PPI when EL1 physical timer expires and PPI must be 30.	CPU System Register Write, GIC APIs	No	Level 2+
203	The local PE timer must generate a PPI when the virtual timer expires, PPI must be 27.	CPU System Register Write, GIC APIs	No	Level 2+
204	The local PE timer must generate a PPI when the EL2 physical timer expires and must be 26.	CPU System Register Write, GIC APIs	No	Level 2+
205	For systems where PE is v8.1 or greater, local PE timer must generate a PPI when the EL2 virtual timer expires and must be 28.	CPU System Register Write, GIC APIs	No	Level 2+
206	In systems that implement EL3, the memory mapped timer (the CNTBaseN frame and associated NTCTLBase frame) must be mapped into the Non-secure address space.	Read/write to Base address	No	Level 2+
206	If the system includes a system Wakeup Timer, this memory-mapped timer must be mapped to Non-secure address space.	Read/write to base address	No	Level 3+
207	Unless all the local PE timers are ON, the base server system implements a system-specific system wakeup timer.	ACPI GTDT	No	Level 1+
208	A system-specific timer generates an SPI.	Platform-specific	No	Level 0+

4.4 Watchdog

Watchdog functionality is verified from running the test on a single PE in the system.

Test ID	Test case	System interface dependency	Requirement of Secure firmware	Level
301	The system implements a Generic Watchdog.	ACPI GTDT	No	Level 1+
-	The watchdog must have both its register frames mapped on to Non-secure address space, which is referred to as the Non-secure watchdog.	-	-	Level 3+
302	Watchdog signal 0 is routed as an SPI to the GIC and usable as an EL2 interrupt.	ACPI GTDT, GIC APIs	No	Level 1+

Test ID	Test case	System interface dependency	Requirement of Secure firmware	Level
-	Watchdog signal 0 is routed as an SPI or LPI to the GIC and usable as an EL2 interrupt.	-	-	Level 2+
303	A system compatible with level 5 will implement a generic counter which counts in nanosecond units. Arm strongly recommends that such systems use revision 1 of the generic watchdog.	ACPI GTDT	No	Level 5+

4.5 PCIe

PCIe functionality is verified from running the test on a single PE in the system.

Test ID	Test case	System interface dependency	Requirement of Secure firmware	Level
401	Systems must map memory space to PCI Express configuration space, using the PCI Express Enhanced Configuration Access Mechanism (ECAM). Tests must be robust to ARI that is implemented.	UEFI PCD, FDT, ACPI	No	Level 1+
402	The base address of each ECAM region is discoverable from system firmware data.	ACPI MCFG table	No	Level 1+
403	PEs can access the ECAM region.	PCI Root Bridge IO Protocol read/write	No	Level 1+
405	All systems support mapping PCI Express memory space as either device memory or Non-cacheable memory. When PCI Express memory space is mapped as normal memory, the system must support unaligned accesses to that region.	Memory map and read/write	No	Level 1+
406	In a system with an SMMU for PCIe, there are no transformations to addresses that the PCIe devices send before they are presented as an input address to the SMMU.	-	-	Level 0+
407	Support for Message Signaled Interrupts (MSI or MSI-X) is required for PCIe devices. MSI and MSI-X are edge-triggered interrupts that are delivered as a memory write transaction.	-	-	Level 1+
408	Each unique MSI or MSI-X will trigger an interrupt with a unique ID and the MSI or MSI-X will target GIC registers requiring no hardware- specific software to service the interrupt.	-	-	Level 1+
409	All MSIs and MSI-X are mapped to LPI.	-	-	Level 2+
410	If the system supports PCIe PASID, then at least 16 bits of PASID must be supported.	-	-	Level 3+

Test ID	Test case	System interface dependency	Requirement of Secure firmware	Level
411	The PCIe Root Complex is in the same Inner Shareable domain as the PEs.	-	-	Level 0+
412	Each of the 4 legacy interrupt lines must be allocated a unique SPI ID and is programmed as level sensitive.	-	-	Level 1+
413	All Non-secure on-chip master in a base Server system that are expected to be under the control of the OS or hypervisor must be capable of addressing all the NS address space. If the master go through an SMMU, then it must address all the NS address space when the SMMU is off. Non-secure off-chip devices that cannot directly address all the Non-secure address space must be placed behind the stage 1 SMMU compatible with the Arm SMMUv2 or SMMUv3 specification. This has an output address size large enough to address all the Non-secure address space.	-	-	Level 3+
414	Memory Attributes of DMA traffic are one of the following: • Inner WB, Outer WB, Inner Shareable • Inner/Outer Non- Cacheable • Device Type IO coherent DMA is as per Inner/Outer WB, Inner Shareable.	-	-	Level 3+
415	PCI Express transactions not marked as No_snoop accessing memory that the PE translation tables attribute as Non-cacheable and shared are I/O coherent with the PEs. I/O coherency fundamentally means that no software coherency management is required on the PEs for the PCI Express root complex, and therefore devices, to get a coherent view of the PE memory. PCI Express transactions marked as No_snoop accessing memory that the PE translation tables attribute as cacheable and shared behave correctly when the appropriate SW coherence is deployed.	-	-	Level 0+
416	For Non-prefetchable (NP) memory, type-1 headers only support 32-bit address, systems complaint with SBSA level 4 or above must support 32-bit programming of NP BARs on such endpoints.	-	-	Level 4+
417	In a system where the PCIe hierarchy allows peer to peer transactions, the Root Ports in an Arm-based SoC must implement PCIe access control service (ACS) features.	-	-	Level 3+
418	All PCIe switches should support the minimal features.	-	-	Level 3+

Test ID	Test case	System interface dependency	Requirement of Secure firmware	Level
419	All multi-function devices, SR- IOV and non-SR-IOV, that are capable of peer-to-peer traffic between different functions should support the minimal features.	-	-	Level 3+
420, 431, 432	All PCIe devices, must implement the common registers of Type 0/1 header.	-	-	Level 3+
421, 434	All PCIe devices, must implement the registers of type 0 header.	-	-	Level 3+
422	All PCIe devices, must implement the registers of type 1 header.	-	-	Level 3+
423	i-EP Root Port must implement the registers of PCIe capability(10h).	-	-	Level 3+
424, 433, 435	All PCIe devices must implement the Device capability register of PCIe capability (10h).	-	-	Level 3+
425	All PCIe devices must implement the Device Control register of PCIe capability(10h).	-	-	Level 3+
426, 436, 437	All PCIe devices must implement the device capabilities 2 register of PCIe capability (10h).	-	-	Level 3+
427	All PCIe devices must implement the device control 2 register of PCIe capability(10h).	-	-	Level 3+
428	All PCIe devices must implement the power management capability register of power management capability(01h).	-	-	Level 3+
429	All PCIe devices must implement the power management control/status register of power management capability(01h).	-	-	Level 3+
430	Memory space access should raise unsupported request when device memory space enable bit is clear	-	-	Level 3+
438 439	iEP root port must follow completion timeout ranges supported, completion time-out disables supported, and AtomicOp routing supported bit.	-	-	Level 3+
440	Root Port must not support ATS and PRS extended capability.	-	-	Level 3+
441	RCiEP and iEP end point must support MSI or MSI-X interrupts.	-	-	Level 3+
442	RCiEP, iEP root port and iEP end point must support power management capability.	-	-	Level 3+
443	Root Port must implement ARI forwarding enable.	-	-	Level 3+

Test ID	Test case	System interface dependency	Requirement of Secure firmware	Level
444	Root Port Configuration Space must be under same ECAM as the Configuration Space of Endpoints and switches in hierarchy that originates from that port.	-	-	Level 3+
445	All Root Port configuration space under same host bridge must be in same ECAM	-	-	Level 3+
446	The Root Port must comply with the byte enable rules and support 1-byte, 2-byte and 4-byte configuration read and write requests.	-	-	Level 3+
447	Recognition and usage of configuration transactions for the Root Port configuration space and read/write the appropriate Root Port configuration register.	-	-	Level 3+
448	Recognition of transactions received on the primary side of the RP PCI-PCI bridge, targeting NP memory spaces of devices and switches that are on the secondary side of the bridge. Address falls within the NP memory window in the type 1 header registers.	-	-	Level 3+
449	Must recognize transactions received on the primary side of the RP PCI-PCI bridge, targeting P memory spaces of devices and switches that are on the secondary side of the bridge: Address falls within the preferential memory window in the type 1 header registers.	-	-	Level 3+
450	Each legacy interrupt SPI must be programmed as level-sensitive in the appropriate GIC_ICFGR.	-	-	Level 3+
451	For i-EP, the Root Port must provide the ability to do a hot reset of the Endpoint using the Secondary Bus Reset bit in bridge Control Register.	-	-	Level 3+
452	PCIe ATS capability must be supported if the RCiEP or i-EP has a software visible cache for address translations.	-	-	Level 3+
453	If the PCIe hierarchy allows peer-to-peer transactions, Root Port must support ACS capability.	-	-	Level 3+
454	If the PCIe hierarchy allows peer-to-peer transactions. The root port must support ACS violation error detection, Logging and reporting must be through the usage of AER mechanism.	-	-	Level 3+
455	If the Root port supports P2P with other root ports and if the root port supports ATS and P2P traffic with other root ports, then it must support ACS direct translated P2P.	-	-	Level 3+

Test ID	Test case	System interface dependency	Requirement of Secure firmware	Level
456	If the i-EP endpoint can send transactions to a peer endpoint (RCiEP or i-EP endpoint or discrete), then the i-EP root port must have ACS capability.	-	-	Level 3+
457	ACS capability must be present in the RCiEP or i-EP endpoint functions if the RCiEP or i-EP Endpoint ISA multi-function device and supports P2P traffic between its functions.	-	-	Level 3+

4.6 Exerciser

Exerciser functionality is verified by running the tests on a single PE in the system.

Test ID	Test case	System interface dependency	Requirement of Secure firmware	Level
801	PEs can access the ECAM region.	-	-	Level 3+
802	PEs can access the BAR address of Exerciser.	-	-	Level 3+
803	In a system where the PCI Express does not use an SMMU, the PCI Express devices have the same view of physical memory as the PEs.	DMA transactions trigger	-	Level 3+
804	Each unique MSI(-X) shall trigger an interrupt with a unique ID and the MSI(-X) shall target GIC registers requiring no hardware- specific software to service the interrupt.	System Interrupt ITS Support, LPI Support, MSI(-X) mapping	-	Level 3+
805	If the system supports PCIe PASID, then at least 16 bits of PASID must be supported.	-	-	Level 3+
806	Trigger Legacy Interrupt using Interrupt Pin register.	System Interrupt and Interrupt Pin mapping	-	Level 3+
807	PCI Express transactions not marked as No_snoop accessing memory that the PE translation tables attribute as cacheable and shared are I/O coherent with the PEs.	DMA transactions trigger	-	Level 3+
808	Memory space access should raise Unsupported Request, when device Memory Space enable bit is clear of RootPort.	-	-	Level 3+
809	Configuration transactions indented for secondary bus of root port must be of Type0.	Platform-specific	-	Level 3+
810	Configuration transactions indented for subordinate bus range of root port must be of type1.	Platform-specific	-	Level 3+
811	Address Translation Service (ATS) functionality check.	DMA transactions trigger, ATS Support	-	Level 3+

Test ID	Test case	System interface dependency	Requirement of Secure firmware	Level
812	Check peer-to-peer ACS source validation and transaction blocking functionality.	DMA transactions trigger, P2P functionality, 2 Exerciser on different Root port with ACS Support	-	Level 3+
813	Check peer-to-peer ACS redirected request validation functionality.	DMA transactions trigger, P2P functionality, 2 Exerciser on different Root port with ACS Support	-	Level 3+
814	PCI Express transactions marked as No_snoop that are accessing memory must have coherency managed by software.	DMA transactions trigger	-	Level 3+
815	Transactions that are targeted at devices must be treated as device-type accesses. They must be ordered and not be merged and allocated in caches.	Platform-specific	-	Level 3+
816	Root Port must implement ARI forwarding enable.	-	-	Level 3+

4.7 Wakeup Semantics

Wakeup semantics functionality is verified from running the test on a single PE in the system.

Test ID	Test case	System interface dependency	Requirement of Secure firmware	Level
501	Wake up from power semantic B due to EL0 Physical Timer Interrupt (PTI).	System Register write, GIC APIs	No	Level 2+
502	Wake up from power semantic B due to EL0 Virtual Timer Interrupt (VTI).	System Register write, GIC APIs	No	Level 2+
503	Wake up from power semantic B due to EL2 PTI.	System Register write, GIC APIs	No	Level 2+
504	Wake up from power semantic B due to watchdog WS0 interrupt.	System Register write, GIC APIs	No	Level 2+
505	Wake up from power semantic B due to system timer interrupt.	Platform code	No	Level 2+

4.8 Peripherals

Peripheral functionality is verified from running the test on a single PE in the system.

Test ID	Test case	System interface dependency	Requirement of Secure firmware	Level
601	If the system has a USB 2.0 host controller peripheral, it must conform to EHCI v1.1 or later. But peripheral subsystems which do not conform to the same are permitted if they are not required to boot and install an OS.	USB EHCIHostController Protocol	No	Level 0+
-	If the system has a USB 3.0 host controller Peripheral, it must conform to XHCI v1.0 or later. But peripheral subsystems which do not conform to the above are permitted if they are not required to boot and install an OS.	USB XHCIHostController Protocol	-	
602	If the system has a SATA host controller peripheral it must conform to AHCI v1.3 or later. But peripheral subsystems which do not conform to the above are permitted if they are not required to boot and install an OS.	SATA AHCIHostController	No	Level 0+
603	To system development and bring up, the base server system will include a Generic UART. The Generic UART is specified in Appendix B. The UARTINTR interrupt output is connected to the GIC as an SPI.	Protocol	No	Level 1+
-	Check that the Generic UART is mapped to Non- secure address space.	Register read	-	Level 3+
604	UARTINTR of the generic UART will be connected as SPI or LPI.	Yes	No	Level 2+
606	Secure generic UART is present. It is not aliased in Non-secure address space. The UARTINTR output of the Secure generic UART is connected to the GIC as an SPI.	Register read/write	Yes	Level 3+

4.9 IO Virtualization (SMMU)

IO Virtualization functionality is verified from running the test on a single PE in the system.

Test ID	Test case	System interface dependency	Requirement of Secure firmware	Level
701	If SMMU is present, then 64KB granule must be supported.	Register read	No	Level 0+
702	All the SMMUs in the system must be compliant with the same architecture version.	ACPI IORT table	No	Level 3+
703	If SMMUv3 is in use, the integration of the System MMUs is compliant with the specification in Appendix H: SMMUv3 Integration.	ACPI IORT table	No	Level 3+

Test ID	Test case	System interface dependency	Requirement of Secure firmware	Level
-	A System MMU compatible with the Arm SMMUv2 or SMMUv3 specification must provide stage 2 System MMU functionality.	Register read	-	
704	The SMMUv3 specification requires that PCIe root complex must not use the stall model due to potential deadlock.	ACPI table, Register read	-	Level 3+
705	If SMMUv2 is in use, each context bank must present a unique physical interrupt to the GIC.	Yes	-	Level 3+
706	Each function, or virtual function, that requires hardware IO Virtualization is associated with an SMMU context. The programming of this association is IMPLEMENTATION DEFINED and is expected to be described by system firmware data.	-	-	Level 1+
707	SMMU version check	ACPI IORT table, Register read	-	Level 1+
708	SMMU must implement support for 16-bit VMID.	ACPI IORT table, Register read	-	Level 6+
709	SMMU must implement support for 16-bit ASID.	ACPI IORT table, Register read	-	Level 6+
710	SMMU must support the translation granule sizes supported by the PEs.	ACPI IORT table, Register read	-	Level 6+
711	If PEs implement Armv8.2-LVA, the SMMU must support extended virtual addresses	ACPI IORT table, Register read	-	Level 6+
712	If PEs implement Armv8.2-LPA, the SMMU must support a 52-bit output size	ACPI IORT table, Register read	-	Level 3+
713	SMMU must implement coherent access to memory structures, queues, and page tables	ACPI IORT table, Register read	-	Level 6+
714	SMMU must support Hardware Translation Table Update (HTTU) of the Access flag and the Dirty state of the page for AArch64 translation tables.	ACPI IORT table, Register read	-	Level 6+
715	SMMU supports little endian for translation table walks, and at a minimum must match the endianness support of the PEs.	ACPI IORT table, Register read	-	Level 6+
716	The DVM capabilities of all DVM receivers (SMMUs and PEs) must be the same or a superset of the DVM capabilities of all DVM initiators (PEs). Check for TLB Range Invalidation.	ACPI IORT table, Register read	-	Level 6+

5 Test Scenarios

The test scenarios are divided based on the functionality and the hardware domain access. The test suite follows this division of test scenarios to better categorize the test report.

The level of target compliance is an input to each of these test scenarios. The scenarios are classified into the following:

5.1 VAL APIs

The following VAL APIs are consumed by all the tests and are not mentioned explicitly for each test.

- val_initialize_test
- val_run_test_payload
- val_pe_get_index_mpid
- val_pe_get_mpid
- val_set_status
- val_report_status

5.2 PE

The VAL API val_pe_create_info_table must be called before any of the following test scenarios are executed.

5.2.1 Number of PEs

The PEs referred to in the SBSA specification are those that are running the operating system or hypervisor, not PEs that are acting as devices.

Does not exceed 2^28

Test ID	VAL APIs consumed	Compliance level applicable
1	val_pe_get_num	Level 2+

5.2.2 PEs must implement SIMD extensions

ID_AA64PFR0_EL1 must indicate support bits [23:20].

Test ID	VAL APIs consumed	Compliance level applicable
2	val_pe_reg_read	Level 0+

5.2.3 PEs must implement 16-bit ASID support

ID_AA64MMFR0_EL1 must indicate support for 16-bit ASIDs in ASIDBits == 0010 for all cores.

Test ID	VAL APIs consumed	Compliance level applicable
3	val_pe_reg_read	Level 0+

5.2.4 PEs must support 4KB and 64KB at stage 1 and 2

ID_AA64MMFR0_EL1 must indicate support for 4KB and 64KB granules for all cores.

Test ID	VAL APIs consumed	Compliance level applicable
3	val_pe_reg_read	Level 0+

5.2.5 Cache are implemented as VIPT or PIPT

CTR_EL0 bits 15:14 must indicate the instruction cache type.

Test ID	VAL APIs consumed	Compliance level applicable
5	val_pe_reg_read	Level 0+

5.2.6 All PEs are coherent and in the same Inner Shareable domain

ID_MMFR0_EL1.InnerShr must indicate hardware coherency support for InnerShr across all cores, ShreLvl must be 0001 across all cores (later is mandated for Armv8). Functional verification is optional.

Test ID	VAL APIs consumed	Compliance level applicable
6	val_pe_reg_read	Level 0+

5.2.7 PEs must implement Cryptography extensions

ID_ISAR5_EL1 must indicate support for SHA1 and SHA2, AES, and PMULL and PMULL2 instructions. This test must be run only when export restriction allows Cryptography Extensions.

Test ID	VAL APIs consumed	Compliance level applicable
7	val_pe_reg_read	Level 0+

5.2.8 PEs must have LE support

ID_AA64MMFR0_EL1 indicates whether mixed-endian support is present. If mixed-endian is not supported, then SCTLR_ELx.EE must strictly read as 0 indicating endianness as little-endian. If mixed-endian is supported, then memory reads with toggled SCTLR_ELx.EE must return swizzled data.

Test ID	VAL APIs consumed	Compliance level applicable
8	val_pe_reg_read	Level 0+

5.2.9 PEs must implement AArch64

ID_AA64PFR0_EL1 must indicate support for AArch64 for all levels.

Test ID	VAL APIs consumed	Compliance level applicable
10	val_pe_reg_read	Level 0+

5.2.10 PMU overflow signal

The generated PMUIRQ must be wired to unique ID and returned as part of the platform code. Must be wired to PPI 23

Test ID	VAL APIs consumed	Compliance level applicable
10	val_pe_reg_read, val_pe_reg_write, val_gic_install_isr, val_pe_get_pmu_gsiv	Level 2+

5.2.11 PMU counters

Implement minimum of 6

Test ID	VAL APIs consumed	Compliance level applicable
12	val_pe_reg_read	Level 1+

5.2.12 PEs must implement a minimum of four synchronous watchpoints

ID_AA64DFR0_EL1.WRPs must indicate a value of at least 3.

Test ID	VAL APIs consumed	Compliance level applicable
13	val_pe_reg_read	Level 0+

5.2.13 Breakpoints

ID_AA64DFR0_EL1.BRPs indicates number of breakpoints implemented.
ID_AA64DFR0_EL1.CTX_CMPs should read at least 1.

Implement minimum of 6. ID_AA64DFR0_EL1.WRPs must indicate a value of at least 5.
ID_AA64DFR0_EL1.CTX_CMPs must read at least 1.

Test ID	VAL APIs consumed	Compliance level applicable
14	val_pe_reg_read	Level 1+

5.2.14 All PEs are architecturally symmetric

Read all the processor ID registers from all PEs and then compare the values with the main PE (cpu_id 0).

Test ID	VAL APIs consumed	Compliance level applicable
15	val_pe_reg_read, val_set_test_data, val_data_cache_ci_va	Level 0+

5.2.15 CRC32 instruction must be implemented

Read processor register ID_AA64ISAR0_EL1 bits 19:16.

Test ID	VAL APIs consumed	Compliance level applicable
17	val_pe_reg_read	Level 3+

5.2.16 If SVE and SPE are implemented, then PEs implement Armv8.3-SPE

Read ID_AA64DFR0_EL1.PMSVer[35:32] = 0b0010 for v8.3-SPE

Test ID	VAL APIs consumed	Compliance level applicable
18	val_pe_reg_read	Level 6+

5.2.17 All PEs must implement the RAS extension introduced in Armv8.2

Read PE register ID_AA64PFR0_EL1 bits 31:28.

Test ID	VAL APIs consumed	Compliance level applicable
19	val_pe_reg_read	Level 4+

5.2.18 All PEs must implement support for 16-bit VMD

Read PE register ID_AA64MMFR1_EL1 bits 7:4.

Test ID	VAL APIs consumed	Compliance level applicable
20	val_pe_reg_read	Level 4+

5.2.19 All PEs must implement virtual host extensions

Read PE register ID_AA64MMFR1_EL1 bits 11:8.

Test ID	VAL APIs consumed	Compliance level applicable
21	val_pe_reg_read	Level 4+

5.2.20 If PEs implement Armv8.3 pointer signing, then they must provide the standard algorithm defined by the Arm architecture

Read PE register ID_AA64ISAR1_EL1 and check bits[7:4], bits[11:8], bits[27:24] and bits[31:28].

Test ID	VAL APIs consumed	Compliance level applicable
22	val_pe_reg_read	Level 4+

5.2.21 All PEs must implement enhanced nested Virtualization

Read PE register ID_AA64MMFR2_EL1.FWB bits 27:24.

Test ID	VAL APIs consumed	Compliance level applicable
23	val_pe_reg_read	Level 5+

5.2.22 All PEs must support changing of page table, mapping size using level 1 and level 2 solution proposed in the Armv8.4 extension

Read PE register ID_AA64MMFR2_EL1.FWB bits 55:52.

Test ID	VAL APIs consumed	Compliance level applicable
24	val_pe_reg_read	Level 5+

5.2.23 All PEs must provide support for stage 2 control of memory types and Cacheability, as introduced by Armv8.4 extensions

Read PE register ID_AA64MMFR2_EL1.FWB bits 43:40.

Test ID	VAL APIs consumed	Compliance level applicable
25	val_pe_reg_read	Level 5+

5.2.24 All PEs must implement the Activity Monitors Extension

Read PE register ID_AA64PFR0_EL1 bits 47:44.

Test ID	VAL APIs consumed	Compliance level applicable
26	val_pe_reg_read	Level 5+

5.2.25 Where export control allows, all PEs must implement cryptography support for SHA3 and SHA512

Read PE register ID_AA64ISAR0_EL1.SHA3 bits [35:32] and bits [15:12].

Test ID	VAL APIs consumed	Compliance level applicable
27	val_pe_reg_read	Level 5+

5.2.26 Hardware updates to Access flag and Dirty state in translation tables, must be supported

Read ID_AA64MMFR1_EL1.HAFDBS[3:0] = 0b0010 For Hardware update supported.

Test ID	VAL APIs consumed	Compliance level applicable
28	val_pe_reg_read	Level 6+

5.2.27 PEs must implement restrictions on speculation introduced in the Armv8.5 extensions

Read PE Register ID_AA64PFR0_EL1.CSV2[59:56] = 0b0010 speculative use of out of Ctxt branch targets and ID_AA64PFR0_EL1.CSV3[63:60] = 0b0001 speculative use of Faulting data.

Test ID	VAL APIs consumed	Compliance level applicable
29	val_pe_reg_read	Level 6+

5.2.28 PEs must implement Speculative Store Bypass Safe

Read PE Register ID_AA64PFR1_EL1.SSBS[7:4] = 0b0010

Test ID	VAL APIs consumed	Compliance level applicable
30	val_pe_reg_read	Level 6+

5.2.29 PEs must implement the SB speculation barrier read

Read PE Register ID_AA64ISAR1_EL1.SPECRES[43:40] = 0b0001

Test ID	VAL APIs consumed	Compliance level applicable
31	val_pe_reg_read	Level 6+

5.2.30 PEs must implement the CFP RCTX, DVP RCTX, CPP RCTX instructions

Read PE Register ID_AA64ISAR1_EL1.SPECRES[43:40] = 0b0001 For CFP, DVP, CPP RCTX instructions.

Test ID	VAL APIs consumed	Compliance level applicable
32	val_pe_reg_read	Level 6+

5.2.31 PEs must provide support for Branch Target Identification

Read PE Register ID_AA64PFR1_EL1.BT[3:0] = 0b0001 For Branch Target Identification Support

Test ID	VAL APIs consumed	Compliance level applicable
33	val_pe_reg_read	Level 6+

5.2.32 PEs must protect against timing faults being used to guess page table mappings

Read ID_AA64MMFR2_EL1.E0PD[63:60] = 0b0001 For Support for Protect Against Timing Fault

Test ID	VAL APIs consumed	Compliance level applicable
34	val_pe_reg_read	Level 6+

5.2.33 PEs provide support for enhanced virtualization traps

Read ID_AA64MMFR2_EL1.EVT[59:56] = 0b0010 - Support for Enhanced Virtualization Trap

Test ID	VAL APIs consumed	Compliance level applicable
35	val_pe_reg_read	Level 6+

5.2.34 All PEs implement Armv8.5-PMU

Read ID_AA64DFR0_EL1.PMUVer[11:8] = 0b0110 For Support for PMU v8.5 Support

Test ID	VAL APIs consumed	Compliance level applicable
36	val_pe_reg_read	Level 6+

5.3 GIC

The VAL API val_gic_create_info_table needs to be called before any of the following test scenarios are executed.

5.3.1 GIC version

GICv2 is implemented. ID registers are at offset 0xFE8 (ICPIDR2.ArchRev) == 0x2. On ACPI tables, GICD structure in MADT must indicate revision 2 for the GIC.

Test ID	VAL APIs consumed	Compliance level applicable
101	val_gic_get_info	Level 0, 1

GIC V3 is implemented

Test ID	VAL APIs consumed	Compliance level applicable
101	val_gic_get_info	Level 2+

5.3.2 If the system includes PCIe, then the GICv3 interrupt controller implements ITS and LPI

Check if ECAM is present, if yes, assume the system implements PCIe. Check for the presence of ITS from MADT table and HW register value.

Test ID	VAL APIs consumed	Compliance level applicable
102	val_gic_get_info, val_pcie_get_info	Level 2+

5.3.3 The GICv3 interrupt controller supports two Security states

Check GICD_CTLR.DS bit (bit6 == 0 : 2 states, bit 6 == 1 : 1 state).

Test ID	VAL APIs consumed	Compliance level applicable
103	val_gic_get_gicd_base, val_gic_get_info	Level 3+

5.3.4 GIC maintenance interrupt is wired as PPI 25

The generated GIC maintenance interrupt must be wired as PPI 25

Test ID	VAL APIs consumed	Compliance level applicable
104	val_gic_get_info, val_gic_install_isr, val_gic_reg_read, val_gic_reg_write, val_gic_end_of_interrupt	Level 3+

5.4 System and Generic Timer

Call the VAL API val_timer_create_info_table before any of the following test scenarios are executed.

5.4.1 System counter of the Generic Timer runs at a minimum frequency of 10 and at a maximum frequency of 400MHz

ACPI GTDT table gives the frequency of the timer. The test must check that the frequency matches the value read from CNTFREQ registers. The functional test of the timer clock frequency is beyond the capability of the AVS suite.

Test ID	VAL APIs consumed	Compliance level applicable
201	val_gic_get_timer_info	Level 0+

5.4.2 The local PE timer when expiring must generate a PPI when the EL1 physical timer expires

This must test the overflow when programming CNTP_TVAL_EL0 or CNTP_CVAL_EL0. The test must ensure for each CPU a PPI is generated, and the PPI is the same for all CPUs. Must be wired to PPI 30

Test ID	VAL APIs consumed	Compliance level applicable
202	val_gic_get_timer_info, val_gic_install_isr val_timer_set_phy_el1	Level 2+

5.4.3 The local PE timer when expiring must generate a PPI when the virtual timer expires

This must test the overflow when programming CNTV_TVAL_EL0 or CNTV_VAL_EL0. The test must ensure for each CPU a PPI is generated, and the PPI is the same for all CPUs. Must be wired to PPI 27

Test ID	VAL APIs consumed	Compliance level applicable
203	val_gic_get_timer_info, val_gic_install_isr val_timer_set_vir_el1	Level 2+

5.4.4 The local PE timer when expiring must generate a PPI when the EL2 physical timer expires

This must test the overflow when programming CNTHP_TVAL_EL2 or CNTHP_CVAL_EL2. The test must ensure for each CPU a PPI is generated, and the PPI is the same for all CPUs. Must be wired to PPI 26

Test ID	VAL APIs consumed	Compliance level applicable
204	val_gic_get_timer_info, val_gic_install_isr val_timer_set_phy_el2	Level 2+

5.4.5 The Local PE timer when expiring must generate a PPI when the EL2 virtual timer expires

Must be wired to a unique PPI for the associated PE. Must be wired to PPI 28

Test ID	VAL APIs consumed	Compliance level applicable
205	val_gic_get_timer_info, val_gic_install_isr val_timer_set_vir_el2, val_pe_reg_read	Level 2+

5.4.6 In systems that implement EL3, the memory mapped timer must be mapped into the Non-secure address space (the CNTBaseN frame and associated CNTCTLBase frame)

Test ID	VAL APIs consumed	Compliance level applicable
206	val_gic_get_timer_info val_mmio_read, val_mmio_write	Level 1+

If the system includes a system wakeup timer, this memory-mapped timer must be mapped on to Non-secure address space

Test ID	VAL APIs consumed	Compliance level applicable
206	val_gic_get_timer_info val_mmio_read val_mmio_write	Level 3+

5.4.7 Unless all the local PE timers are always on, the base server system implements a system-specific system wakeup timer

Test ID	VAL APIs consumed	Compliance level applicable
207	val_gic_get_timer_info	Level 0+

5.4.8 System-specific system timer generates an SPI

Test ID	VAL APIs consumed	Compliance level applicable
208	val_timer_get_info, val_timer_skip_if_cntbase_access_not_allowed val_gic_install_isr, val_timer_set_system_timer, val_timer_disable_system_timer, val_gic_end_of_interrupt	Level 0+

5.5 Watchdog

Call the VAL API val_wd_create_info_table before any of the following test scenarios are executed.

5.5.1 System implements a Generic Watchdog as specified in SBSA specification

Test ID	VAL APIs consumed	Compliance level applicable
301	val_wd_get_info	Level 1+

The Non-secure watchdog must have both its register frames mapped on to Non-secure address space

Test ID	VAL APIs consumed	Compliance level applicable
301	val_wd_get_info, val_mmio_read	Level 2+

5.5.2 Watchdog Signal 0 is routed as SPI (or LPI) and usable as an EL2 interrupt

WS0 routed as SPI

Test ID	VAL APIs consumed	Compliance level applicable
302	val_wd_get_info, val_gic_install_isr val_wd_set_ws0	Level 0, 1

WS0 routed as SPI or LPI

Test ID	VAL APIs consumed	Compliance level applicable
302	val_wd_get_info, val_gic_install_isr, val_wd_set_ws0	Level 2, 3

5.5.3 A system compatible with level 5 implements a generic counter which counts in nanosecond units.

Test ID	VAL APIs consumed	Compliance level applicable
303	val_wd_get_info	Level 5+

5.6 Peripherals and Memory

Call the VAL APIs val_peripheral_create_info_table, and val_memory_create_info_table for relevant test scenarios before their execution

5.6.1 If the system has a USB2.0 (USB3.0) host controller peripheral, it must conform to EHCI v1.1 (XHCI v1.0) or later

Test ID	VAL APIs consumed	Compliance level applicable
601	val_peripheral_get_info, val_pcie_read_cfg	Level 0+

5.6.2 If the system has a SATA host controller peripheral, it must conform to AHCI v1.3 or later

Test ID	VAL APIs consumed	Compliance level applicable
602	val_peripheral_get_info, val_pcie_read_cfg	Level 0+

5.6.3 Base server system includes a Generic UART

Test ID	VAL APIs consumed	Compliance level applicable
603	val_peripheral_get_info	Level 0+

5.6.4 The UARTINTR interrupt output is connected to the GIC.

UARTINTR routed as SPI or LPI

Test ID	VAL APIs consumed	Compliance level applicable
604	val_peripheral_get_info, val_gic_install_isr	Level 1+

5.6.5 Non-secure access to Secure address must cause exception

Some memory is mapped in secure address space. The memory shall not be aliased in Non-secure address space.

Test ID	VAL APIs consumed	Compliance level applicable
606	val_pe_install_esr, val_pe_update_elr val_pe_reg_read	Level 3+

5.7 Power states and wakeup

There is no prerequisite VAL APIs for the following tests.

5.7.1 In state B, a PE must be able to wake on receipt of an SGI, PPI or SPI that directly targets the PE

Wake up due to EL0 PTI

Test ID	VAL APIs consumed	Compliance level applicable
501	val_timer_get_info val_timer_set_phy_el1 val_gic_install_isr, val_power_enter_semantic	Level 0+

Wake up due to EL0 VTI

Test ID	VAL APIs consumed	Compliance level applicable
502	val_timer_get_info val_timer_set_vir_el1 val_timer_set_phy_el1 val_gic_install_isr, val_power_enter_semantic	Level 0+

Wake up due to EL2 PTI

Test ID	VAL APIs consumed	Compliance level applicable
503	val_timer_get_info val_timer_set_phy_el2 val_timer_set_phy_el2 val_gic_install_isr, val_power_enter_semantic	Level 0+

Wake up due to Watchdog WS0 Interrupt

Test ID	VAL APIs consumed	Compliance level applicable
504	val_wd_get_info val_wd_set_ws0 val_timer_get_info val_timer_set_phy_el1 val_gic_install_isr, val_power_enter_semantic	Level 0+

Wake up due to system time interrupt

Test ID	VAL APIs consumed	Compliance level applicable
505	val_timer_get_info, val_timer_set_system_timer, val_gic_install_isr, val_power_enter_semantic	Level 0+

5.8 IO Virtualization

The VAL API val_smmu_create_info_table needs to be called before any of the following test scenarios are executed.

5.8.1 SMMU if present is compatible with Arm SMMU v1

This test case can be skipped as it is very unlikely that the 2016/2017 platforms will have an SMMU compatible with version 1.

5.8.2 SMMU if present, must support a 64KB translation granule

ID register gives the supported translation granule size.

Test ID	VAL APIs consumed	Compliance level applicable
701	val_smmu_get_info, val_smmu_read_cfg	Level 0+

5.8.3 All the System MMUs in the system must be compliant with the same architecture version

Test ID	VAL APIs consumed	Compliance level applicable
702	val_smmu_get_info	Level 3+

5.8.4 If PCIe, check the stall model

Test ID	VAL APIs consumed	Compliance level applicable
704	val_smmu_get_info, val_pcie_get_info	Level 3+

5.8.5 If SMMUv3 is in use, check the compliance with Appendix E: SMMUv3 integration

Test ID	VAL APIs consumed	Compliance level applicable
703	val_smmu_get_info, val_smmu_read_cfg	Level 3+

5.8.6 If SMMUv2 is in use, each context bank must present a unique physical interrupt to the GIC

Test ID	VAL APIs consumed	Compliance level applicable
705	val_smmu_get_info, val_iovirt_check_unique_ctx_i ntid	Level 3+

5.8.7 Each function, or virtual function, that requires hardware I/O Virtualization is associated with an SMMU context

The programming of this association is IMPLEMENTATION DEFINED and is expected to be described by system firmware data.

Test ID	VAL APIs consumed	Compliance level applicable
706	val_smmu_get_info, val_iovirt_unique_rid_strid_m ap	Level 3+

5.8.8 SMMU Version Check

Test ID	VAL APIs consumed	Compliance level applicable
707	val_smmu_get_info()	Level 1+

5.8.9 SMMU must implement support for 16-bit VMID

Read SMMUv3_IDR0 register to check support for 16-bit VMID

Test ID	VAL APIs consumed	Compliance level applicable
708	val_pcie_get_info(), val_smmu_get_info()	Level 6+

5.8.10 SMMU must implement support for 16-bit ASID

Read SMMUv3_IDR0[12] register to check support for 16-bit ASID

Test ID	VAL APIs consumed	Compliance level applicable
709	val_pcie_get_info(), val_smmu_get_info()	Level 6+

5.8.11 SMMU must support the translation granule sizes supported by the PEs.

Read SMMUv3_IDR5 for granule support in SMMU, and ID_AA64MMFR0_EL1 for granule support in PEs.

Test ID	VAL APIs consumed	Compliance level applicable
710	val_pcie_get_info(), val_smmu_get_info(), val_pe_reg_read(), val_smmu_read_cfg()	Level 6+

5.8.12 If PEs implement Armv8.2-LVA, the SMMU must support extended virtual addresses

Read ID_AA64MMFR2_EL1 [19:16] for Armv8.2-LVA support, then check SMMU_IDR5.VAX = 0b01.

Test ID	VAL APIs consumed	Compliance level applicable
711	val_pcie_get_info(), val_smmu_get_info()	Level 6+

5.8.13 If PEs implement Armv8.2-LPA, SMMU must support a 52 bit output size

Read ID_AA64MMFR0_EL1 [3:0] for Armv8.2-LPA support, then check SMMU_IDR5.OAS = 0b110

Test ID	VAL APIs consumed	Compliance level applicable
712	val_pcie_get_info(), val_smmu_get_info()	Level 3+

5.8.14 The SMMU must implement coherent access to memory structures, queues, and page tables

Read SMMU Register SMMU_IDR0.COHACC[4] == 1 for support.

Test ID	VAL APIs consumed	Compliance level applicable
713	val_pcie_get_info(), val_smmu_get_info()	Level 6+

5.8.15 The SMMU must support HTTU of the Access flag and the Dirty state of the page for AArch64 translation tables

Read SMMU Register SMMU_IDR0.HTTU[7:6] == 0b10

Test ID	VAL APIs consumed	Compliance level applicable
714	val_pcie_get_info(), val_smmu_get_info()	Level 6+

5.8.16 SMMU supports little endian for translation table walks, and at a minimum must match the endianness support of the PEs

Check SCTLR_ELx for endianness support in PEs and SMMUv3_IDR0[22:21] for endianness support in SMMU.

Test ID	VAL APIs consumed	Compliance level applicable
715	val_pcie_get_info(), val_smmu_get_info()	Level 6+

5.8.17 The DVM capabilities of all DVM receivers (SMMUs and PEs) must be the same or a superset of the DVM capabilities of all DVM initiators (PEs). Check for TLB Range Invalidation

Check ID_AA64ISAR0_EL1[59:56] == 0x2 for TLB Range invalidation support in PE's, then check for SMMU_IDR3.RIL = 0b1.

Test ID	VAL APIs consumed	Compliance level applicable
716	val_pcie_get_info(), val_smmu_get_info()	Level 6+

5.9 PCIE

Call the VAL API val_pcie_create_info_table before any of the following test scenarios are executed.

5.9.1 Systems must map memory space to PCI Express configuration space, using the PCI Express Enhanced Configuration Access Mechanism (ECAM)

Test ID	VAL APIs consumed	Compliance level applicable
401	val_pcie_get_info	Level 1

5.9.2 ECAM value present in MCFG

Test ID	VAL APIs consumed	Compliance level applicable
402	val_pcie_get_info	Level 1+

5.9.3 PEs can access ECAM

Test ID	VAL APIs consumed	Compliance level applicable
403	val_pcie_get_info, val_mmio_read	Level 1+

5.9.4 PCIe space is device or non-cacheable

Test ID	VAL APIs consumed	Compliance level applicable
405	val_pcie_get_info, val_memory_get_info	Level 1+

5.9.5 In a system with an SMMU for PCI Express there are no transformations to addresses being sent by PCI Express devices before they are presented as an input address to the SMMU.

The addresses sent by PCI Express devices must be presented to the memory system or SMMU unmodified.

Test ID	VAL APIs consumed	Compliance level applicable
406	val_pcie_get_info val_memory_get_info, val_dma_get_info val_smmu_ops, val_dma_device_get_dma_addr, val_dma_mem_alloc	Level 1+

5.9.6 Support for Message Signaled Interrupts (MSI/MSI-X) is required for PCI Express devices.

MSI and MSI-X are edge-triggered interrupts that are delivered as a memory write transaction.

Test ID	VAL APIs consumed	Compliance level applicable
407	val_peripheral_get_info, val_pcie_get_device_type	Level 1+

5.9.7 Each unique MSI(-X) shall trigger an interrupt with a unique ID and the MSI(-X) shall target GIC registers requiring no hardware-specific software to service the interrupt.

Test ID	VAL APIs consumed	Compliance level applicable
408	val_peripheral_get_info, val_get_msi_vectors	Level 1+

5.9.8 All MSIs and MSI-x are mapped to LPI.

Test ID	VAL APIs consumed	Compliance level applicable
409	val_peripheral_get_info, val_get_msi_vectors	Level 1+

5.9.9 If the system supports PCIe PASID, then at least 16 bits of PASID must be supported

Test ID	VAL APIs consumed	Compliance level applicable
410	val_peripheral_get_info, val_smmu_get_info, val_smmu_max_pasids	Level 1+

5.9.10 The PCI Express root complex is in the same Inner Shareable domain as the PEs

Test ID	VAL APIs consumed	Compliance level applicable
411	val_iovirt_get_pcie_rc_info	Level 1+

5.9.11 Each of the 4 legacy interrupt lines must be allocated a unique SPI ID and is programmed as level sensitive

Test ID	VAL APIs consumed	Compliance level applicable
412	val_peripheral_get_info, val_pci_get_legacy_irq_map	Level 1+

5.9.12 All Non-secure on-chip masters in a base server system that are expected to be under the control of the OS or hypervisor must be capable of addressing all the NS address space.

If the master goes through an SMMU then it must be capable of addressing all of the NS address space when the SMMU is off. Non-secure off-chip devices that cannot directly address all of the Non-secure address space must be placed behind a stage 1 System MMU compatible with the Arm SMMUv2 or SMMUv3 specification that has an output address size large enough to address all of the Non-secure address space.

Test ID	VAL APIs consumed	Compliance level applicable
413	val_peripheral_get_info, val_pcie_is_devicedma_64bit, val_pcie_is_device_behind_smmu	Level 1+

5.9.13 Memory Attributes of DMA traffic

Memory Attributes of DMA traffic are one of (1) Inner WB, Outer WB, Inner Shareable (2) Inner/Outer Non- Cacheable (3) Device type IO coherent DMA is as per (1) Inner/Outer WB, Inner Shareable.

Test ID	VAL APIs consumed	Compliance level applicable
414	val_dma_get_info val_dma_mem_alloc, val_dma_mem_get_attrs	Level 1+

5.9.14 PCI Express transactions not marked as No_snoop accessing memory that the PE translation tables attribute as cacheable and shared are I/O Coherent with the PEs.

Test ID	VAL APIs consumed	Compliance level applicable
415	val_peripheral_get_info, val_pcie_get_device_type, val_pcie_get_dma_support, val_pcie_get_snoop_bit	Level 1+

5.9.15 For Non-prefetchable (NP) memory, type-1 headers only support 32-bit address, systems complaint with SBSA level 4 or above must support 32-bit programming of NP BARs on such endpoints

Test ID	VAL APIs consumed	Compliance level applicable
416	val_peripheral_get_info, val_pcie_get_device_type, val_pcie_io_read_cfg, val_pcie_scan_bridge_devices_and_check_memtype	Level 3+

5.9.16 Root Port must implement minimal ACS features if P2P supported

Test ID	VAL APIs consumed	Compliance level applicable
417	val_peripheral_get_info() val_pcie_get_pcie_type() val_pcie_p2p_support() val_pcie_read_ext_cap_word()	Level 3+

5.9.17 All switches must implement minimal ACS features if P2P supported

Test ID	VAL APIs consumed	Compliance level applicable
418	val_peripheral_get_info() val_pcie_get_pcie_type() val_pcie_p2p_support(), val_pcie_read_ext_cap_word()	Level 3+

5.9.18 Multifunction devices must implement minimal ACS features if P2P supported

Test ID	VAL APIs consumed	Compliance level applicable
419	val_peripheral_get_info(), val_pcie_get_pcie_type(), val_pcie_multifunction_suppo rt() val_pcie_p2p_support(), val_pcie_read_ext_cap_word()	Level 3+

5.9.19 Type 0/1 common config rules check

Test ID	VAL APIs consumed	Compliance level applicable
420	val_pcie_register_bitfields_ch eck(), val_pcie_disable_eru(bdf), val_pcie_device_port_type(bdf) val_pcie_bitfield_check(), val_pcie_find_capability	Level 3+

5.9.20 Type 0 config header rules check

Test ID	VAL APIs consumed	Compliance level applicable
421	val_pcie_register_bitfields_check(), val_pcie_disable_eru(bdf) val_pcie_device_port_type(bdf) val_pcie_bitfield_check(), val_pcie_find_capability	Level 3+

5.9.21 Type 1 config header rules check

Test ID	VAL APIs consumed	Compliance level applicable
422	val_pcie_register_bitfields_check(), val_pcie_disable_eru(bdf), val_pcie_device_port_type(bdf) val_pcie_bitfield_check(), val_pcie_find_capability	Level 3+

5.9.22 PCIe capability rules check

Test ID	VAL APIs consumed	Compliance level applicable
423	val_pcie_register_bitfields_check() val_pcie_disable_eru(bdf) val_pcie_device_port_type(bdf) val_pcie_bitfield_check() val_pcie_find_capability	Level 3+

5.9.23 Device capabilities register rules check

Test ID	VAL APIs consumed	Compliance level applicable
424	val_pcie_register_bitfields_check() val_pcie_disable_eru(bdf) val_pcie_device_port_type(bdf) val_pcie_bitfield_check() val_pcie_find_capability	Level 3+

5.9.24 Device Control register rule check

Test ID	VAL APIs consumed	Compliance level applicable
425	val_pcie_register_bitfields_check() val_pcie_disable_eru(bdf) val_pcie_device_port_type(bdf) val_pcie_bitfield_check() val_pcie_find_capability	Level 3+

5.9.25 Device capabilities 2 register rules check

Test ID	VAL APIs consumed	Compliance level applicable
426	val_pcie_register_bitfields_check() val_pcie_disable_eru(bdf) val_pcie_device_port_type(bdf) val_pcie_bitfield_check() val_pcie_find_capability	Level 3+

5.9.26 Device control 2 reg rules check

Test ID	VAL APIs consumed	Compliance level applicable
427	val_pcie_register_bitfields_check(), val_pcie_disable_eru(bdf) val_pcie_device_port_type(bdf) val_pcie_bitfield_check(), val_pcie_find_capability	Level 3+

5.9.27 Power management capability rules check

Test ID	VAL APIs consumed	Compliance level applicable
428	val_pcie_register_bitfields_check() val_pcie_disable_eru(bdf) val_pcie_device_port_type(bdf) val_pcie_bitfield_check() val_pcie_find_capability	Level 3+

5.9.28 Power management/status rule check

Test ID	VAL APIs consumed	Compliance level applicable
429	val_pcie_register_bitfields_check(), val_pcie_disable_eru(bdf) val_pcie_device_port_type(bdf) val_pcie_bitfield_check() val_pcie_find_capability	Level 3+

5.9.29 Check Command Register memory space enable functionality

Test ID	VAL APIs consumed	Compliance level applicable
430	val_pe_update_elr() val_pcie_bdf_table_ptr() val_pe_install_esr() val_pcie_function_header_typ e() val_pcie_get_downstream_fu nction() val_pcie_get_mmio_bar() val_pcie_disable_eru() val_pcie_clear_urd() val_pcie_disable_msa() val_mmio_read() val_pcie_is_urd() val_pcie_enable_msa() val_pcie_find_capability()	Level 3+

5.9.30 Type0/1 BIST Reg verification rule

Test ID	VAL APIs consumed	Compliance level applicable
431	val_pcie_bdf_table_ptr()	Level 3+

5.9.31 Check HDR CapPtr Reg verification rule

Test ID	VAL APIs consumed	Compliance level applicable
432	val_pcie_bdf_table_ptr()	Level 3+

5.9.32 Max payload size supported check

Test ID	VAL APIs consumed	Compliance level applicable
433	val_pcie_bdf_table_ptr() val_pcie_find_capability()	Level 3+

5.9.33 BAR memory space and type rule check

Test ID	VAL APIs consumed	Compliance level applicable
434	val_pcie_bdf_table_ptr() val_pcie_device_port_type() val_pcie_is_onchip_peripheral() val_pcie_find_capability()	Level 3+

5.9.34 Function level reset rule check

Test ID	VAL APIs consumed	Compliance level applicable
435	val_pe_get_index_mpid() val_pcie_bdf_table_ptr() val_pcie_device_port_type(bdf) val_pcie_find_capability() val_memory_alloc() val_pcie_get_bdf_config_addr() val_memcpy() val_time_delay_ms() val_memory_free() val_pcie_is_onchip_peripheral()	Level 3+

5.9.35 Check ARI forwarding support rule

Test ID	VAL APIs consumed	Compliance level applicable
436	val_pcie_bdf_table_ptr() val_pcie_device_port_type(bdf) val_pcie_find_capability() val_pcie_is_onchip_peripheral()	Level 3+

5.9.36 Check OBFF supported rule

Test ID	VAL APIs consumed	Compliance level applicable
437	val_pcie_bdf_table_ptr() val_pcie_device_port_type(bdf) val_pcie_find_capability() val_pcie_is_onchip_peripheral()	Level 3+

5.9.37 Check CTRS and CTDS rule

Test ID	VAL APIs consumed	Compliance level applicable
438	val_pcie_bdf_table_ptr() val_pcie_device_port_type(bdf) val_pcie_find_capability() val_pcie_is_onchip_peripheral() val_pcie_get_rp_transaction_frwd_su pport(bdf)	Level 3+

5.9.38 Check i-EP atomicop rule

Test ID	VAL APIs consumed	Compliance level applicable
439	val_pcie_bdf_table_ptr() val_pcie_device_port_type(bdf) val_pcie_find_capability() val_pcie_get_atomicop_requester_capable() val_pcie_is_onchip_peripheral()	Level 3+

5.9.39 Check Root Port ATS and PRI rule

Test ID	VAL APIs consumed	Compliance level applicable
440	val_pcie_bdf_table_ptr() val_pcie_device_port_type(bdf) val_pcie_find_capability() val_pcie_is_onchip_peripheral()	Level 3+

5.9.40 Check MSI and MSI-X support rule

Test ID	VAL APIs consumed	Compliance level applicable
441	val_pcie_bdf_table_ptr(), val_pcie_device_port_type(bdf), val_pcie_find_capability(), val_pcie_is_onchip_peripheral()	Level 3+

5.9.41 Check Power Management rules

Test ID	VAL APIs consumed	Compliance level applicable
442	val_pcie_bdf_table_ptr() val_pcie_device_port_type(bdf) val_pcie_find_capability() val_pcie_is_onchip_peripheral()	Level 3+

5.9.42 Check ARI forwarding enable rule

Test ID	VAL APIs consumed	Compliance level applicable
443	val_pcie_bdf_table_ptr() val_pcie_device_port_type(bdf) val_pcie_find_capability() val_pcie_is_onchip_peripheral()	Level 3+

5.9.43 Check device under RP in same ECAM

Test ID	VAL APIs consumed	Compliance level applicable
444	val_pcie_bdf_table_ptr() val_pcie_device_port_type(bdf) val_pcie_get_ecam_base(bdf) val_mmio_read() val_pcie_io_read_cfg() val_pcie_get_info()	Level 3+

5.9.44 Check all RP under a HB is in same ECAM

Test ID	VAL APIs consumed	Compliance level applicable
445	val_pcie_bdf_table_ptr() val_pcie_device_port_type(bdf) val_pcie_get_info() val_pcie_get_ecam_base(bdf) val_pcie_find_capability() val_pcie_is_onchip_peripheral()	Level 3+

5.9.45 The Root port must comply with the byte enable rules and must support 1 byte, 2 byte and 4-byte Configuration read and write

Test ID	VAL APIs consumed	Compliance level applicable
446	val_pcie_bdf_table_ptr() val_pcie_device_port_type(bdf) val_pcie_get_info() val_pcie_get_ecam_base(bdf) val_mmio_read() val_mmio_read8() val_mmio_read16() val_mmio_write() val_mmio_write8() val_mmio_write16()	Level 3+

5.9.46 Recognition and consumption configuration transactions intended for the Root Port configuration space and read/write the appropriate Root Port Configuration register

Test ID	VAL APIs consumed	Compliance level applicable
447	val_pcie_bdf_table_ptr() val_pcie_device_port_type(bdf) val_pcie_get_ecam_base(bdf) val_mmio_read() val_pcie_io_read_cfg()	L3+

5.9.47 Recognition of transactions received on the primary side of the RP PCI-PCI bridge, targeting non-prefetchable memory spaces of devices and switches that are on the secondary side of the bridge. Where the address falls within the non-prefetchable memory window in the type 1 header registers, the transactions must be forwarded to the secondary side

Test ID	VAL APIs consumed	Compliance level applicable
448	val_pcie_bdf_table_ptr() val_pe_install_esr() val_pcie_enable_bme(bdf) val_pcie_enable_msa(bdf) val_pcie_device_port_type(bdf) val_pcie_clear_urd(bdf) val_pcie_read_cfg()	L3+

5.9.48 Recognition of transactions received on the primary side of the RP PCI-PCI bridge, targeting prefetchable memory spaces of devices and switches that are on the secondary side of the bridge. Where the address falls within the prefetchable memory window in the type 1 header registers, the transactions must be forwarded to the secondary side

Test ID	VAL APIs consumed	Compliance level applicable
449	val_pcie_bdf_table_ptr() val_pe_install_esr() val_pcie_enable_bme(bdf) val_pcie_enable_msa(bdf) val_pcie_device_port_type(bdf) val_pcie_clear_urd(bdf) val_pcie_read_cfg()	L3+

5.9.49 Each legacy interrupt SPI must be programmed as level-sensitive in the appropriate GIC_ICFGR

Test ID	VAL APIs consumed	Compliance level applicable
450	val_memory_alloc() val_pcie_bdf_table_ptr() val_pcie_read_cfg() val_pci_get_legacy_irq_map() val_gic_get_intr_trigger_type()	L3+

5.9.50 For i-EP, the Root port must provide the ability to do a hot reset of the Endpoint using the Secondary Bus Reset bit in bridge Control Register

Test ID	VAL APIs consumed	Compliance level applicable
451	val_pcie_bdf_table_ptr(), val_pcie_device_port_type(bdf), val_memory_alloc(), val_pcie_get_bdf_config_addr(), val_memcpy(), val_pcie_read_cfg(), val_pcie_write_cfg(), val_memory_free()	L3+

5.9.51 PCIe ATS capability must be supported if the RCiEP or i-EP has a software visible cache for address translations

Test ID	VAL APIs consumed	Compliance level applicable
452	val_pcie_bdf_table_ptr(), val_pcie_device_port_type(bdf), val_pcie_is_host_bridge(bdf), val_pcie_is_cache_present(bdf), val_pcie_find_capability()	L3+

5.9.52 If the PCIe hierarchy allows peer-to-peer transactions, Root Port must support ACS capability

Test ID	VAL APIs consumed	Compliance level applicable
453	val_pcie_p2p_support() val_pcie_bdf_table_ptr() val_pcie_device_port_type(bdf) val_pcie_find_capability() val_pcie_read_cfg()	L3+

5.9.53 If the PCIe hierarchy allows peer-to-peer transactions, the root port must support ACS violation error detection, Logging and reporting must be through the usage of AER mechanism

Test ID	VAL APIs consumed	Compliance level applicable
454	val_pcie_p2p_support() val_pcie_bdf_table_ptr() val_pcie_device_port_type(bdf) val_pcie_find_capability()	L3+

5.9.54 If the Root port supports peer-to-peer traffic with other root ports then - If the root port supports Address Translation services and peer-to-peer traffic with other root ports, then it must support ACS direct translated P2P

Test ID	VAL APIs consumed	Compliance level applicable
455	val_pcie_bdf_table_ptr() val_pcie_device_port_type(bdf) val_pcie_dev_p2p_support() val_pcie_find_capability() val_pcie_read_cfg()	L3+

5.9.55 If the i-EP endpoint can send transactions to a peer endpoint (RCiEP or i-EP endpoint or discrete), then the i-EP root port must have ACS capability

Test ID	VAL APIs consumed	Compliance level applicable
456	val_pcie_p2p_support(), val_pcie_bdf_table_ptr(), val_pcie_device_port_type(bdf), val_pcie_dev_p2p_support(), val_pcie_get_rootport(), val_pcie_find_capability(), val_pcie_read_cfg()	L3+

5.9.56 ACS capability must be present in the RCiEP or i-EP endpoint functions if the RCiEP or i-EP Endpoint ISA multi-function device and supports peer to peer traffic between its functions.

Test ID	VAL APIs consumed	Compliance level applicable
457	val_pcie_bdf_table_ptr() val_pcie_device_port_type(bdf) val_pcie_multifunction_support() val_pcie_find_capability() val_pcie_read_cfg()	L3+

Appendix A Revisions

This appendix describes the technical changes between released issues of this book.

Table A-1 Issue 02

Change	Location
First release	-

Table A-2 Difference between Issue 02 to Issue 03

Change	Location
New PCIe tests are added.	See PCIE

Table A-3 Difference between Issue 03 to Issue 04

Change	Location
Added SBSA Level 6 PE and SMMU tests.	See • PE • IO Virtualization
Added new PCIe tests and Exerciser tests that are related to Address Translation Service, Peer-to-Peer, and ACS rules.	See PCIE

Table A-4 Difference between Issue 04 to Issue 05

Change	Location
Memory test for unpopulated address space access waived off and implemented for bare-metal.	See Test Scenarios
Fixes in test PE and GIC.	See • PE • GIC
Enabling mmio prints dumps on demand.	See Test Scenarios
Enabled new ARI test	See Test Scenarios
Enabled bare-metal driver for GIC.	See GIC
PCIe Enumeration enhancements.	See PCIE
Bug fix and enhancements related to P2P, SMMU and PCIe.	See • IO Virtualization • PCIE
Updated interrupt related test cases.	See Test Scenarios
Additional support provided for running ACS on bare-metal.	See Test Scenarios