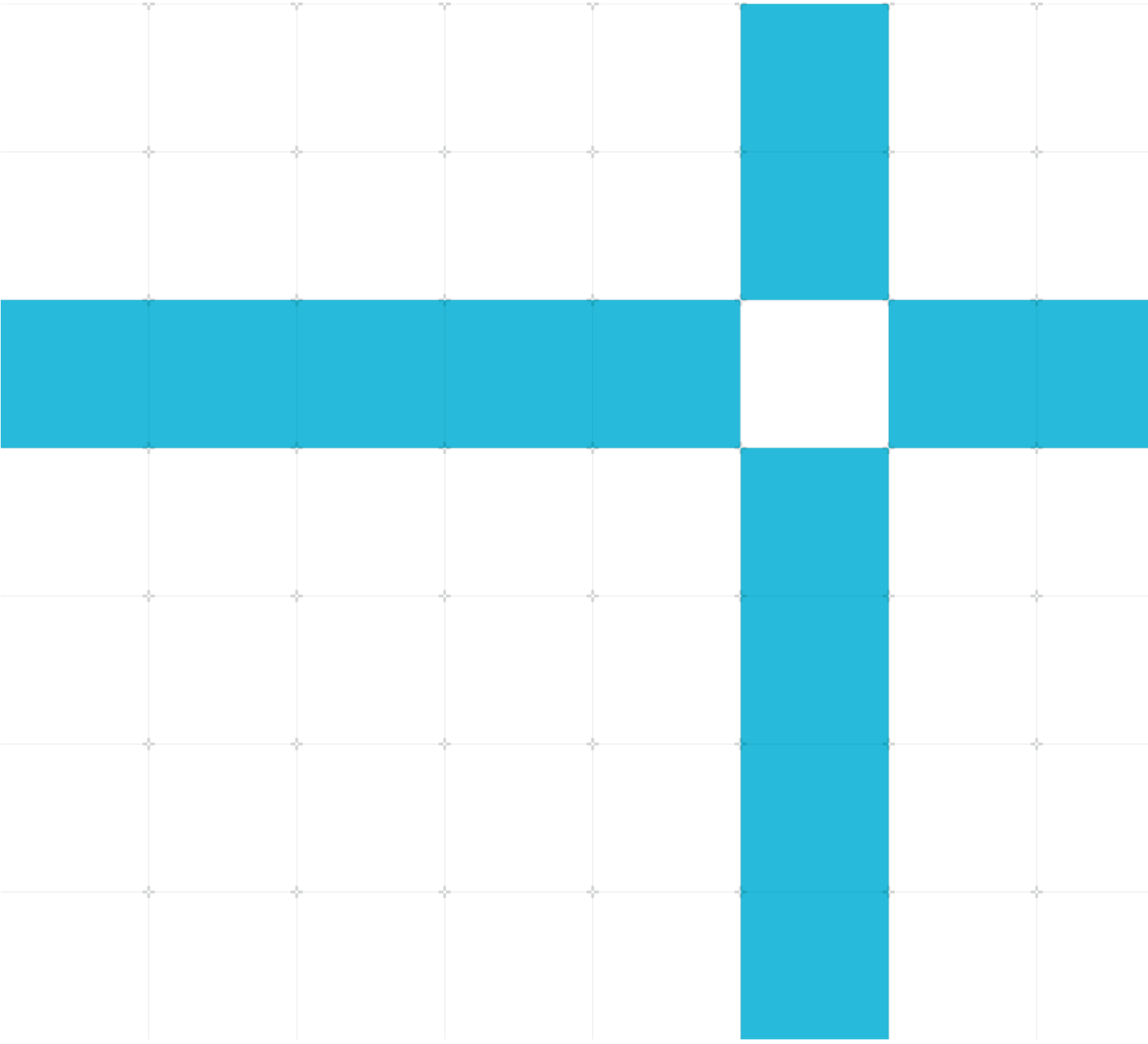
Arm® PC BSA Architecture Compliance

Revision: r1p8

Test Scenario

Non-Confidential

Issue 0108
ARM040-1254092399-18632



Arm PC BSA Test Scenario Document

Release information

Document history

Issue	Date	Confidentiality	Change
01	11 December 2024	Non-Confidential	Alpha release 0.5.0
02	30 July 2025	Non-Confidential	Beta release 0.8.0

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 Beta product.

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on [Product Name], 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 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*mpn* identifier indicates the revision status of the product described in this book, for example, r*1*p*2*, where:

rm Identifies the major revision of the product, for example, r*1*.

pn 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® PC 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 DEFINED, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.

1.4 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® PC Base System Architecture 1.0	DEN0151	Non-Confidential

Arm architecture and specifications	Document ID	Confidentiality
Arm® Architecture Reference Manual for A-profile architecture	DDI0487F	Non-Confidential

Non-Arm resources	Document ID	Organization
PCI Express Base Specification Revision 5.0, Version 1.0	NA	PCI-SIG
PCI-To-PCI Bridge Architecture Specification 1.2	NA	PCI-SIG

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**.

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 Personal Computing Base System Architecture Scenario.
- The number ARM040-1254092399-18632.
- 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 Personal Computing Base System Architecture

The PC Base System Architecture (PC-BSA) specifies a standard hardware system architecture for Personal Computers (PCs) that are based on the Arm 64-bit Architecture. PC system software, for example operating systems, hypervisors, and firmware can rely on this standard system architecture. PC-BSA extends the requirements specified in the Arm BSA.

Personal Computing Base System Architecture (PC BSA) specifies a hardware system architecture based on Arm 64-bit architecture, that system software such as operating systems, hypervisors, and firmware can rely on. The document addresses PE features and key aspects of system architecture. The primary goal of the document is to ensure sufficient standard system architecture to enable a suitably built single OS image to run on all the hardware compliant with the specifications.

Arm does not mandate compliance with this specification. However, Arm anticipates that OEMs, ODMs, cloud service providers, and software providers will require compliance to maximize Out-of-Box software compatibility and reliability.

2.1 PC-BSA ACS

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 a hierarchy is consistent with the target hardware subsystem which is validated by a test. A test may check for different parameters of the hardware subsystem.

The tests are classified as:

- PE
- Memory Map
- GIC
- SMMU
- PCIe
- Watchdog
- Platform Security
- TPM

Test number	Rule ID	Level	Scenario	Algorithm
-	P_L1_01	L1	The base PC system must comply with BSA Level 1 requirements as specified by BSA Level 1 checklist section from Arm BSA [3].	Compliance to this rule requires BSA ACS to be run, refer PC BSA README.md for instructions.

2.2 PE

Test number	Rule ID	Level	Scenario	Algorithm
4, 18	P_L1PE_01	L1	PEs must support 4KB translation granules at stage 1 and stage 2.	ID_AA64MMFR0_EL1 must indicate support for 4KB granules for all cores.
24	P_L1PE_02	L1	PEs must implement 16-bit ASID support.	CPU system register field ID_AA64MMFR0_EL. ASIDBits must read b0010.
25	P_L1PE_03	L1	PEs must implement AArch64 at all implemented Exception levels.	CPU system register ID_AA64PFR0_EL1 fields EL0, EL1, EL2 and EL3 must read non-zero value.
15	P_L1PE_04	L1	PEs must comply with FEAT_LSE requirements as specified by B_PE_25 from Arm BSA	CPU system register ID_AA64ISAR0_EL1.Atomic = 0b0010 or 0b0011
26	P_L1PE_05	L1	Base PC systems that make use of FEAT_LPA must support a functional system memory map which is wholly contained within the address range from 0 to 2^48-1	Check for FEAT_LPA presence, CPU System register ID_AA64MMFR0_EL1 bits [3:0] must read b0110. If FEAT_LPA is implemented, then ID_AA64MMFR0_EL1.TGran16 [23:20] must read 0b0010 or ID_AA64MMFR0_EL1.TGran4 [31:28] must read 0b001. If FEAT_LPA2 is also implemented along with FEAT_LPA, then all peripheral addresses have no restriction else all peripheral addresses should be contained inside 2^48 memory map.
28	P_L1PE_06	L1	If the system contains persistent memory that is exposed to the OS, all PEs must support the clean to point of persistence instruction (DC CVAP). The instruction must be able to perform a clean to the point of persistence for all memory that is exposed as persistent memory to the OS	Check whether persistent memory exists in the system from using UEFI GetMemoryMap() service. If present, CPU System register field ID_AA64ISAR1_EL1.DPB must read b0001 or b0010 indicating DC CVAP instruction support.
29	P_L1PE_07	L1	All PEs must implement FEAT_VMID16	CPU system register field ID_ AA64MMFR1_EL1.VMIDBits must read b0010.
30	P_L1PE_08	L1	All PEs must implement FEAT_VHE.	CPU system register field ID_ AA64MMFR1_EL1.VH must read b0001

2.3 Memory Map

Test number	Rule ID	Level	Scenario	Algorithm
105	P_L1MM_01	L1	If a base PC system supports 64KB translation granules at stage 2 then it must ensure that: 1. All memory and peripherals can be mapped using 64KB stage 2 pages and must not require the use of 4KB pages at stage 2. 2. All peripherals that can be assigned to different virtual machines will be situated within different 64KB regions of memory. This rule applies to types of peripherals such as PCIe devices. Here is an indicative list of the peripherals that are exempt from this rule: • On-chip peripherals whose resources are not managed by PE software, for example system controllers or power management controllers. • Secure-world peripherals.	ID_AA64MMFR0_EL1 must indicate support for 64KB granules for all cores. If yes, then check whether all peripheral base addresses are 64KB apart from each other.

2.4 GIC

Test number	Rule ID	Level	Scenario	Algorithm
212	P_L1GI_01	L1	A base PC system must implement at least one interrupt controller that is compliant with the GICv3 or higher architecture	Check GIC version is 3. or greater than 3
846	P_L1GI_02	L1	All MSI and MSI-X targeting hypervisor and operating system software must be mapped to LPI.	Pull each discovered PCI device and its list of MSI(X) vectors and Check whether every vector IRQ number is an LPI or not.
210	P_L1GI_03	L1	A GIC that implements Locality-specific Peripheral Interrupts (LPIs) should support clearing GICR_CTLR.EnableLPIs	Get ITS Address for current ITS. Check GITS_CTLR.Enabled = 0 and GITS_CTLR. Quiescent = 1

Test number	Rule ID	Level	Scenario	Algorithm
211	P_L1GI_04	L1	A GIC implementation that does not support clearing GICR_CTLR.EnableLPIs after it is set must not permit modification of GICR_PENDBASER when GICR_CTLR.EnableLPIs == 1.	Get RDBase Address for current PE, Check GICR_CTLR.EnableLPIs = 0 and GICR_CTLR.RWP = 0
214	P_L1PP_01	L1	The Interrupt IDs must be the same as the recommended values specified by B_PPI_01, B_PPI_02, and B_PPI_03 from Arm BSA	Check for the reserved interrupt as mentioned in the interrupt table

2.5 SMMU

Test number	Rule ID	Level	Scenario	Algorithm
	P_L1SM_01	L1	If PEs that are used by the base PC system support TLB range instructions, then all OS visible requesters that contain a TLB must support range invalidates. See FEAT_TLBIRANGE in [1].	Not Implemented Which requester are OS visible and have TLB in a system are IMPDEF and no generic way to obtain that information
308	P_L1SM_02	L1	base PC system must support Stage 1 System MMU functionality. This must be provided by a System MMU that is compliant with the Arm SMMUv3, or higher, architecture revision.	The SMMU memory mapped register field SMMU_AIDR.ArchMajorRev ≥ 3 indicating SMMU is compliant with SMMUv3 or higher. And memory mapped register field SMMU_IDR0.S1P must read b1 indicating Stage 1 support
308	P_L1SM_03	L1	A base PC system must support Stage 2 System MMU functionality. This must be provided by a System MMU that is compliant with the Arm SMMUv3, or higher, architecture revision	The SMMU memory mapped register field SMMU_AIDR.ArchMajorRev ≥ 3 indicating SMMU is compliant with SMMUv3 or higher. And memory mapped register field SMMU_IDR0.S2P must read b1 indicating Stage 2 support.
320	P_L1SM_04	L1	The integration of the System MMUs must be compliant with rules SMMU_01 and SMMU_02, as specified in “SMMUv3 integration” section from Arm BSA	The SMMU memory mapped register field SMMU_IDR0.COHACC must read b1.

Test number	Rule ID	Level	Scenario	Algorithm
322	P_L1SM_06	L1	All DMA capable requesters in a system must pass through SMMU translation, including both Secure and Non-secure devices. The requirement does not apply to DMA capable requesters which are part of the TCB of the system.	All PCIe root ports and devices named components in IORT ACPI table should be behind an SMMU, if there are DMA capable. 1.DMA capability of PCIe root ports and named components can be determined by reading CCA attribute field of node in IORT ACPI table. 2. Device is said to be behind an SMMU if output reference in IORT node's ID mapping points to an SMMU.

2.6 PCIe

Test number	Rule ID	Level	Scenario	Algorithm
	P_L1PCI_1	L1	All devices that are intended for assignment to a virtual machine are required to be PCIe compliant.	Not Implemented Manual Verification All peripherals identified in ACPI SPCR table must have bus number non-zero
887	P_L1PCI_2	L1	There must be no OS observable use of PCIe Enhanced Allocation	Pass if No EA Capability is present or check if "Enable" bit in Entry Type register is 0
-	P_L1NV_01	L1	Non-volatile storage must be available that can be accessed and updated directly by firmware without the aid of the operating system.	Ensure UEFI firmware exposes writable non-volatile variables. Validate update capability without OS intervention. Covered by VariableServicesTest SCT testcase. Refer PC BSA README.md for instruction on running SCT.

2.7 Watchdog

Test number	Rule ID	Level	Scenario	Algorithm
701, 702	P_L2WD_01	FR	The base PC system must implement a Non-secure Generic watchdog as specified by B_WD_01, B_WD_02, B_WD_03, B_WD_04, and B_WD_05 in Arm BSA [3].	Read and verify watchdog refresh and control registers 1. Generate the watchdog interrupt. 2. Check if the interruption is reaching GIC. 3. Check the system-specific interrupt controller with interrupt ID Not implemented: B_WD_04, B_wd_05 WS1 signal is routed to a higher privilege entity to perform IMPDEF behavior WS1 signal performs platform-specific action. This is IMPDEF behavior

2.8 Platform Security

Test number	Rule ID	Level	Scenario	Algorithm
	P_L1SE_01	L1	The non-volatile storage for firmware code and protected data (such as UEFI authenticated variables) must be protected from direct modification from a PE running at non-secure privilege levels. Firmware images and protected data must only be modifiable through authenticated interfaces.	Verify that non-volatile storage used by firmware (e.g., UEFI authenticated variables) is not writable by software running at non-secure privilege levels. Attempt direct access/modification from a PE at Non-secure EL; it must be denied. Confirm that updates to protected data occur only through authenticated firmware interfaces. Covered by VariableServicesTest SCT testcase. Refer PC BSA README.md for instruction on running SCT.

Test number	Rule ID	Level	Scenario	Algorithm
	P_L1SE_02	L1	Verified boot must be rooted in an on-chip immutable root-of-trust for verification that authenticates the first mutable code using digital signatures. Authentication means: · Verifying the integrity of the image through a digital signature check · Verifying that the image signing public key is rooted to a trusted authority	Not Implemented Requires manual verification of immutable RoT via SoC documentation or vendor confirmation; cannot be validated programmatically.
	P_L1SE_03	L1	Hashes and asymmetric keys must have a minimum 128-bit security strength (NIST SP 800-57 [7]).	Not Implemented Applies to hardware-fused keys used in verified boot, not UEFI Secure Boot or TPM keys; not testable by ACS.
	P_L1SE_04	L1	A base PC system must have a hardware-backed IMPLEMENTATION DEFINED mechanism that is the equivalent of OTP memory to reflect the security-relevant state of the system.	Not Implemented Mechanism is vendor-specific with no standard interface; presence and behavior must be verified through platform documentation.
	P_L1SE_05	L1	The state described in P_L1SE_04 must be readable by firmware to determine whether a system is in a secure operational/deployed/ production state.	Not Implemented No standard method for firmware to read system state; typically stored in inaccessible fuses or lifecycle controllers.

2.9 TPM

Test number	Rule ID	Level	Scenario	Algorithm
1601	P_L1TP_01	L1	A system must have a TPM implementation that is compliant with the TCG PC Client Platform TPM Profile Specification for TPM 2.0 [8].	1.Check for the presence of the TCG2 UEFI protocol. 2. If available, use the SubmitCommand() API to query the TPM. 3. Parse the response to determine the TPM version (e.g., 1.2 or 2.0).

Test number	Rule ID	Level	Scenario	Algorithm
	P_L1TP_02	L1	It must not be possible to reset a TPM or SoC independently of each other.	Not Implemented TPM reset behavior is hardware-defined and typically tied to SoC reset; no standard software interface exists to verify independent reset capability.
1602	P_L1TP_03	L1	The platform interface to access the TPM must support localities 0 - 4 of the TPM.	Parse the TPM2 ACPI table to get the TPM interface type via StartMethod. For FIFO: • Use AddressOfControlArea (default to 0xFED40000 if 0). • Read TPM_INTERFACE_ID_0 → check CapLocality bit. For CRB: • Read TPM_CRB_INTF_ID_0 → check CapLocality bit. Pass if CapLocality == 1 (supports Localities 0–4); else Fail. Other interfaces (e.g., via SMC/FFA) → Fail (Locality 0 only)
	P_L1TP_04	L1	The system must support hardware-based enforcement to protect locality 4 of the TPM, and it must not be possible for locality 4 to be accessed by Non-secure privilege levels.	Not Implemented Not covered by UEFI-SCT or SBBR Linux tests; no software-accessible method exists to validate secure access restrictions for TPM Locality 4. Requires manual verification.

Appendix A Revisions

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

Table A-1 Issue 01

Change	Location
Initial Release	-

Table A-2 Difference between Issue 01 to Issue 02

Change	Location
Added new module TPM tests and Watchdog tests	See • **TPM** • **Watchdog**
Added New Tests in GIC and PCIe	See • **GIC** • **PCIe**
Added Platform security module	See **Platform Security**