arm

Arm® SBSA Architecture Compliance

Revision: r7p1

User Guide

Arm® SBSA Architecture Compliance
User Guide

Release Information

Document history

Issue	Date	Confidentiality	Change
0701-07	28 May 2025	Non-Confidential	REL 7.2.2 EAC release
0701-06	29 October 2024	Non-Confidential	REL 7.2.1 EAC release
0701-05	29 March 2024	Non-Confidential	REL 7.2.0 BETA release
0701-04	28 September 2023	Non-Confidential	REL 7.1.3 EAC release
0701-03	29 June 2023	Non-Confidential	REL 7.1.2 EAC release
0701-02	28 March 2023	Non-Confidential	REL 7.1.1 BETA-1 release
0701-01	16 January 2023	Non-Confidential	REL 7.1 BETA-0 release
0700-01	15 June 2022	Non-Confidential	REL 7.0 ALPHA release
0601-01	28 October 2022	Non-Confidential	REL 6.1
0302-01	26 July 2022	Non-Confidential	REL 3.2
0301-01	27 September 2021	Non-Confidential	REL 3.1
0300-01	30 September 2020	Non-Confidential	REL 3.0
0200-04	20 March 2020	Non-Confidential	REL 2.4

Issue	Date	Confidentiality	Change
0200-03	18 September 2019	Non-Confidential	REL 2.3
0200-02	26 April 2019	Non-Confidential	REL 2.2
0200-01	27 December 2018	Non-Confidential	REL 2.1. The document now follows a new numbering format.
D	11 May 2018	Non-Confidential	REL 2.0
C	13 July 2017	Non-Confidential	REL 1.0
B	31 March 2017	Non-Confidential	Beta release
A	30 November 2016	Non-Confidential	Alpha release

Proprietary Notice

This document is protected by copyright and other related rights and the use or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm Limited ("Arm"). No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether the subject matter of this document infringes any third party patents.

The content of this document is informational only. Any solutions presented herein are subject to changing conditions, information, scope, and data. This document was produced using reasonable efforts based on information available as of the date of issue of this document. The scope of information in this document may exceed that which Arm is required to provide, and such additional information is merely intended to further assist the recipient and does not represent Arm's view of the scope of its obligations. You acknowledge and agree that you possess the necessary expertise in system security and functional safety and that you shall be solely responsible for compliance with all legal, regulatory, safety and security related requirements concerning your products, notwithstanding any information or support that may be provided by Arm herein. In addition, you are responsible for any applications which are used in conjunction with any Arm technology described in this document, and to minimize risks, adequate design and operating safeguards should be provided for by you.

Confidentiality Status

Unrestricted Access is an Arm internal classification.

Product Status

The information in this document is Final, that is for a developed product.

Feedback on content

Information about how to give feedback on the content.

If you have comments on content then send an e-mail to support-systemready-acs@arm.com.. Give:

- The title Arm® SBSA Architecture Compliance User Guide.

- The number 101547_0701_07_en.

- If applicable, the page number(s) to which your comments refer.

- A concise explanation of your comments.

Arm also welcomes general suggestions for additions and improvements.

> **Note**
> Arm tests the PDF only in Adobe Acrobat and Acrobat Reader, and cannot guarantee the quality of the represented document when used with any other PDF reader.

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 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.

Convention	Use
italic	Citations.
bold	Terms in descriptive lists, where appropriate.
`monospace`	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
`monospace` <u>`underline`</u>	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	Terms that have specific technical meanings as defined in the *Arm® Glossary*. For example, **IMPLEMENTATION DEFINED**, **IMPLEMENTATION SPECIFIC**, **UNKNOWN**, and **UNPREDICTABLE**.

⚠️ **Caution**
We recommend the following. If you do not follow these recommendations your system might not work.

⚠️ **Warning**
Your system requires the following. If you do not follow these requirements your system will not work.

⚠️ **Danger**
You are at risk of causing permanent damage to your system or your equipment, or of harming yourself.

Note	This information is important and needs your attention.

Tip	This information might help you perform a task in an easier, better, or faster way.

Remember	This information reminds you of something important relating to the current content.

1.2 Useful resources

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

Arm documents are available on developer.arm.com/documentation. Confidential documents are only available to licensees, when logged in.

Each document link in the tables below provides direct access to the online version of the document.

Arm product resources	Document ID	Confidentiality
Arm® Server Base System Architecture 7.1	DEN0029H	Non-Confidential
GICv3 and GICv4 Software Overview	DAI0492	Non-Confidential

Arm architecture and specifications	Document ID	Confidentiality
Arm® Architecture Reference Manual for A-profile architecture	DDI0487I.a	Non-Confidential
Arm® Generic Interrupt Controller Architecture Specification for GIC architecture version 3.0 and version 4.0	IHI0069H	Non-Confidential

Note	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.

1.3 Other information

See the Arm® website for other relevant information.

- Arm® Developer.
- Arm® Documentation.
- Technical Support.
- Arm® Glossary.

2. Overview of the SBSA test suites

This chapter provides an overview of the SBSA test suites and the test IDs.

2.1 Abbreviations

The following table lists the abbreviations used in this document.

Table 2-1: Abbreviations and expansions

Abbreviation	Expansion
ACPI	Advanced Configuration and Power Interface
ACS	Architecture Compliance Suite
AEST	Arm Error Source Table
BDF	Bus, Device, and Function
CATU	CoreSight Address Translation Unit
ETE	Embedded Trace Extension
ELx	Exception Level x (where x can be 0 to 3)
ETR	Embedded Trace Router
GIC	Generic Interrupt Controller
GCD	Grand Central Dispatch
HMAT	Heterogeneous Memory Attribute Table
HVC	HyperVisor Call
IOMMU	Input-Output Memory Management Unit
ITS	Interrupt Translation Service
LPI	Locality-specific Peripheral Interrupt
MPAM	Memory System Resource Partitioning and Monitoring
MSI	Message-Signaled Interrupt
MTE	Memory Tagging Extension
NUMA	Non-Uniform Memory Access
PAL	Platform Abstraction Layer
PCIe	Peripheral Component Interconnect express
PCCT	Platform Communications Channel Table
APMT	Performance Monitoring Unit Table
PE	Processing Element
PMU	Performance Monitor Unit
PPTT	Processor Properties Topology Table
PSCI	Power State Coordination Interface
RAS	Reliability, Availability, and Serviceability
RAS2	Reliability, Availability, and Serviceability 2

Abbreviation	Expansion
RCiEP	Root Complex integrated End Point
SATA	Serial Advanced Technology Attachment
SBSA	Server Base System Architecture
SMC	Secure Monitor Call
SMMU	System Memory Management Unit
SRAT	System Resource Affinity Table
STS	Statistical Test Suite
SoC	System on Chip
UEFI	Unified Extensible Firmware Interface
UART	Universal Asynchronous Receiver and Transmitter
VAL	Validation Abstraction Layer

2.2 Overview of tests

The following table describes the general divisions of Server Base System Architecture (SBSA) tests between Unified Extensible Firmware Interface (UEFI) shell application, Linux application, and Bare-metal.

Table 2-2: Test environment and modules

Test environment	Modules
UEFI Shell	PE, GIC, Timers, Watchdog, Wakeup, PCIe, NIST, Peripherals, SMMU, PMU, MPAM, RAS, Memory, and ETE
Linux command line	PCIe, SMMU, PMU
Bare-metal	Exerciser

2.3 Test IDs

Each test ID is generated as an addition of module ID and unit test ID. For a given module, unit test ID begins from 1.

The following table lists the module names and their IDs.

Table 2-3: Module names and module IDs

Module name	Module ID
PE	0
Memory	100
GIC	200
SMMU	300
Timer	400
Power and Wakeup	500

Module name	Module ID
Peripheral	600
Watchdog	700
PCIe	800
Exerciser	900
MPAM	1000
PMU	1100
RAS	1200
NIST	1300
ETE	1400

3. UEFI shell application

This chapter provides information on executing tests from the UEFI Shell application and its PAL API implementation.

3.1 UEFI application arguments

Run the UEFI Shell application with the following set of arguments:

- ```
uefi shell> sbsa.efi [-v <n>] [-l <n>] [-only] [-skip <x,y,z, ..>] [-f <file
  name>] [-nist] [-mmio] [-p2p] [-cache] [-only] [-t <x, y, z>] [-m <x,y,z>] [-
  timeout <wakeup test timeout multiple>] [-fr] [-no_crypto_ext]
```

- ```
uefi shell> Acs.efi [-v <n>] [-skip <x,y,z, ..>] [-f <file name>]
-t <x,y,z> [-m <x,y,z>] [-mmio] [-sbsa] [-timeout <wakeup test timeout multiple>]
  [- p2p] [-cache] [-el1physkip] [-l <n>] [-only] [-fr]
```

> **Note**
>
> By default, only BSA tests run. To run SBSA tests, pass the `-sbsa` parameter.

The following table provides descriptions to the arguments.

Table 3-1: Descriptions of UEFI application arguments

Argument	Description
`-v`	Print level **1** INFO and above. **2** DEBUG and above. **3** TEST and above. **4** WARN and ERROR. **5** ERROR.
`-l`	Level of compliance to be tested for (0-7). The default value is 4.
`-only`	Runs tests of a selected level. This option needs to be passed along with `-l` (level) option.
`-skip`	Overrides the suite to skip the execution of a particular test(s) and/or module(s). For example, 301 skips test case with ID = 301. 500 skips all tests in module with ID = 500. For more information on module IDs, see 2.3 Test IDs on page 11.
`-f`	File name to which the output log is written.
`-nist`	Runs the SBSA ACS with NIST STS.

Argument	Description
`-mmio`	Enables all the mmio read/write prints. **Note:** • To enable pal_mmio_read or write prints, use with `-v 1`. • Enables prints from specific module by using module id. For example, `-mmio 200`, enables for GIC module, and `-mmio 0`, enables for PE module.
`-p2p`	Enables p2p feature support.
`-cache`	Enables cache support.
`-t`	To run only multiple selected tests.
`-m`	To run only multiple selected modules. **Note:** `-m` will override `-t` if used on the same module.
`- only`	To only run tests belonging to a specific level of compliance. -l (level) or -fr option needs to be specified for using this flag.
`-timeout`	Timeout value for wakeup test.
`-fr`	To run SBSA future requirements tests. **Note:** `-l` will override `-fr` if given together.
`-no_crypto_ext`	Pass this flag if cryptography extension is not supported due to export restrictions.

Note

The UEFI session becomes unusable after the SBSA tests are run and the test results are printed on the UEFI console.

Example 1

```
shell> sbsa.efi -v 2 -l 3 -skip 100,36 -f acs.txt
```

The set of parameters shown in the code block:

- Prints messages with verbosity of 2 and above.

- Tests for compliance against SBSA level 3.

- Skips execution of all tests belonging to Memory module and test number 36.

- Stores the log messages to the file `acs.txt`.

Example 2

```
shell > sbsa.efi -m 0,200 -skip 2
```

The set of parameters shown in the code block:

- Runs only the PE and GIC modules.
- Skips the PE test 02.

3.2 UEFI implementation of PAL APIs

This section provides information on infrastructure APIs and module-specific APIs.

Infrastructure APIs

The following table describes the Platform Abstraction Layer (PAL) APIs and UEFI interfaces.

Table 3-2: PAL APIs and UEFI interfaces

PAL API	UEFI interfaces
pal_print	AsciiPrint
mem_alloc	gBS->AllocatePool
mem_free	gBS->FreePool
mem_alloc_shared	gBS->AllocatePool
mem_free_shared	gBS->FreePool
mem_get_shared_addr	None
mem_alloc_cachebale	gBS->AllocatePages
mem_free_cacheable	gBS->FreePages
time_delay_ms	gBS->Stall
mem_alloc_pages	gBS->AllocatePages
mem_free_pages	gBS->FreePages
mmio_read	None
mmio_write	None
aligned_alloc	gBS->AllocatePool
mem_free_aligned	gBS->FreePool

Module-specific APIs

The following table represents the mapping of PAL API to Advanced Configuration and Power Interface (ACPI), if the system firmware presents platform configuration through ACPI tables.

Table 3-3: PAL APIs, UEFI interfaces, and ACPI tables consumed

PAL API	UEFI interfaces consumed	ACPI table consumed
pe_create_info_table	• gST->ConfigurationTable • CompareGuid • IndustryStandard/Acpi.h	MADT Table
pe_execute_payload	-	-
pe_install_esr	• gEfiCpuArchProtocolGuid • Cpu->RegisterInterruptHandler	-

PAL API	UEFI interfaces consumed	ACPI table consumed
gic_create_info_table	• gST->ConfigurationTable • CompareGuid • IndustryStandard/Acpi.h	MADT table
gic_install_isr	• gHardwareInterruptProtocolGuid • RegisterInterruptSource • EnableInterruptSource	-
timer_create_info_table	• gST->ConfigurationTable • CompareGuid • IndustryStandard/Acpi.h	GTDT table
wd_create_info_table	• gST->ConfigurationTable • CompareGuid • IndustryStandard/Acpi.h	GTDT table
pcie_create_info_table	• gST->ConfigurationTable • CompareGuid • IndustryStandard/Acpi.h	MCFG table
pcie_get_mcfg_ecam	• gST->ConfigurationTable • CompareGuid, IndustryStandard/Acpi.h • IndustryStandard/MemoryMappedConfigurationSpaceAccessTable.h	MCFG table
iovirt_create_info_table	• gST->ConfigurationTable • CompareGuid • IndustryStandard/Acpi.h	IORT table
peripheral_create_info_table	• gEfiPciIoProtocolGuid • Pci->GetLocation • Pci->Pci.Read	-
memory_create_info_table	gBS->GetMemoryMap	-
cache_create_info_table	• IndustryStandard/Acpi.h • gST->ConfigurationTable • CompareGuid	PPTT table
hmat_create_info_table	• IndustryStandard/Acpi.h • gST->ConfigurationTable • CompareGuid	HMAT table
ras_create_info_table	• gST->ConfigurationTable • CompareGuid • IndustryStandard/ArmErrorSourceTable.h	AEST table
mpam_create_info_table	• gST->ConfigurationTable • CompareGuid • IndustryStandard/Acpi.h	MPAM table
pmu_create_info_table	• gST->ConfigurationTable • CompareGuid • IndustryStandard/Acpi.h	APMT table

PAL API	UEFI interfaces consumed	ACPI table consumed
ras2_create_info_table	• gST->ConfigurationTable • CompareGuid • IndustryStandard/Acpi.h	RAS2 and PCCT table
srat_create_info_table	• gST->ConfigurationTable • CompareGuid • IndustryStandard/Acpi.h	SRAT table
pcc_create_info_table	• gst->ConfigureTable • CompareGuid • IndustryStandard/Acpi.h	PCCT table

4. Linux application

This chapter provides information on executing tests from the Linux application.

4.1 Linux application arguments

Run the Linux application with the following set of arguments:

```
shell> sbsa [-v <n>] [-l <n>] [--skip <x,y,z>]
```

Table 4-1: Description of Linux application arguments

Argument	Description
-v	Print level 1 INFO and above 2 DEBUG and above 3 TEST and above 4 WARN and ERROR 5 ERROR
-l	Level of compliance to be tested for. (0 to 7)
--skip	Overrides the suite to skip the execution of a particular tests (Upto 3). For example, 53 skips test case with ID 53.

Example

```
shell> sbsa -v 3 -l 3 --skip 57
```

This set of parameters tests for compliance against SBSA level 3 with print verbosity set to 3, and skips test number 57.

Loading the kernel module

Before the SBSA ACS Linux application is run, load the SBSA ACS kernel module using the `insmod` command.

```
shell> insmod sbsa_acs.ko
```

4.2 Environment setup

This section details the target and runtime environment setup.

4.2.1 Test requirements

The set of tests assumes that at least one Serial Advanced Technology Attachment (SATA) controller is behind a PCIe root complex. The SATA controller may or may not be behind an Input-Output Memory Management Unit (IOMMU).

Before running these tests, at least one SATA hard disk must be connected to the SATA controller. The test performs read and write operations to the SATA hard disk. Therefore, the data on the HDD is overwritten. The SATA drive must not be the boot device for the OS.

4.2.2 Runtime environment

The following figure describes the hardware functional blocks.

Figure 4-1: Hardware functional blocks
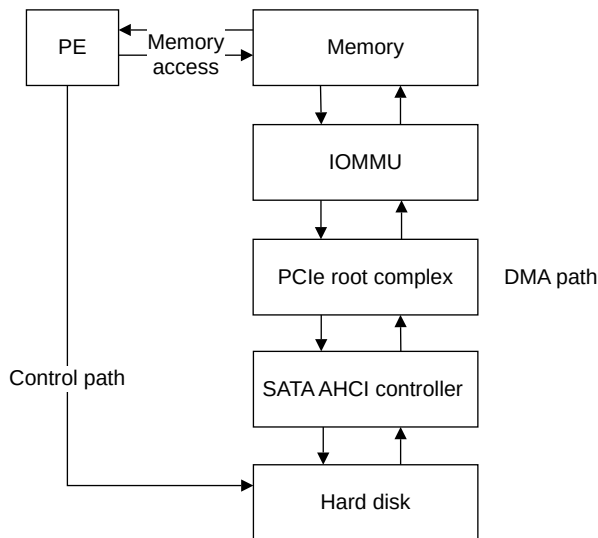
The PCIe-DMA tests initiate data transfers from a DMA requester. By default, the test searches for a SATA controller which is part of the PCIe subsystem.

1. The test programs the known data from the PE to main memory.

2. The test programs the DMA requester to transfer this known data to its end-point device.

3. The test programs the DMA requester to transfer the data back to a different location in the main memory.

4. The test compares the data at both the locations.

If the SATA controller is not placed before an IOMMU, then during this data transfer, the address that is used by the SATA controller is retrieved and compared with the DMA address that is seen by the PE.

If the DMA requester is placed before an IOMMU, then the address that is used by the SATA AHCI controller is compared with the address that is seen by the IOMMU. Both these addresses must match.

To enable the export of the addresses that are seen by the SATA AHCI controller and IOMMU, the kernel drivers for these two modules must be patched.

5. PMU Linux application

This chapter describes how to run the PMU Linux application.

5.1 Running the PMU application

The following steps describe how to run the PMU application.

To run the application, follow these steps:

1. Export the path from which the Python modules are present.

   ```
   export PYTHONPATH=/lib/python3.10/site-packages/
   ```

2. Navigate to the directory where the scripts are present.

   ```
   cd /bin/pmuval
   ```

3. Run script.

   ```
   python sbsa_acs_pmu.py -a
   ```

   Option -a to run on all PEs.

Appendix A Revisions

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

A.1 Revisions

The following tables describe the changes between different issues of this document.

Table A-1: Issue 0200-01

Change	Location
Information about exerciser is added.	See, 2.3 Test IDs on page 11.
A new parameter [--e] is added to Linux application arguments.	See, 4.1 Linux application arguments on page 18.

Table A-2: Differences between Issue 0200-01 and Issue 0200-02

Change	Location
Bare-metal test environment is added to the table.	See, 2.2 Overview of tests on page 11.
A note about additional porting for the exerciser is added.	See, 4.1 Linux application arguments on page 18.

Table A-3: Differences between Issue 0200-02 and Issue 0200-03

Change	Location
No technical changes.	-

Table A-4: Differences between Issue 0200-03 and Issue 0200-04

Change	Location
• Arguments for NIST and PCIe tests are added. • A note about UEFI session is added.	See, 3.1 UEFI application arguments on page 13.
NIST module ID is updated.	See, 2.3 Test IDs on page 11.
Linux application arguments are updated.	See, 4.1 Linux application arguments on page 18.

Table A-5: Differences between Issue 0200-04 and Issue 0300-01

Change	Location
Additional level of compliance to be tested is added.	See table in 3.1 UEFI application arguments on page 13 and 4.1 Linux application arguments on page 18.

Table A-6: Differences between Issue 0300-01 and Issue 0301-01

Change	Location
Removed Secure module.	See, 2.3 Test IDs on page 11.
Updated the link to linux-acs.	See, 4.2 Environment setup on page 18.
Updated the build steps and environment setup.	See, 4.2 Environment setup on page 18.

Table A-7: Differences between Issue 0301-01 and Issue 0302-01

Change	Location
Arguments for p2p and cache are added.	See, 3.1 UEFI application arguments on page 13.

Table A-8: Differences between Issue 0302-01 and Issue 0601-01

Change	Location
Added an abbreviation for HVC.	See, 2.1 Abbreviations on page 10.
Removed Exerciser module from the Linux command line.	See, 2.2 Overview of tests on page 11.
Added new argument options for test id, module id, and timeout with more examples.	See, 3.1 UEFI application arguments on page 13.
Section on Build steps and environment setup is moved to the README file.	-

Table A-9: Differences between Issue 0601-01 and Issue 0700-01

Change	Location
Added information on PMU Linux application	See, 5. PMU Linux application on page 21.
Added new terms in Abbreviations and new APIs in UEFI implementation of PAL APIs.	See, 2.1 Abbreviations on page 10, 3.2 UEFI implementation of PAL APIs on page 15.
Added details for PMU, MPAM, and RAS; removed Exerciser module.	See, 2.2 Overview of tests on page 11.
Updated the arguments and description for UEFI application.	See, 3.1 UEFI application arguments on page 13.
Added new module names and IDs.	See, 2.3 Test IDs on page 11.
Added new PAL APIs.	See, 3.2 UEFI implementation of PAL APIs on page 15.
Updated the commands and versions.	See, 4.2 Environment setup on page 18.

Table A-10: Differences between Issue 0700-01 and Issue 0701-01

Change	Location
Added details for Memory module.	See, 2.2 Overview of tests on page 11.
Updated the arguments and description for UEFI application.	See, 3.1 UEFI application arguments on page 13.
Added a new module Memory.	See, 2.3 Test IDs on page 11.
Added RAS and SRAT APIs.	See, 3.2 UEFI implementation of PAL APIs on page 15.
Added new terms in Abbreviations.	See, 2.1 Abbreviations on page 10.
Updated the steps to run the PMU application.	See, 5.1 Running the PMU application on page 21.

Table A-11: Differences between Issue 0701-01 and Issue 0701-02

Change	Location
Changed the order of modules executed in the Table 2-3: Test environment and modules.	See, 2.2 Overview of tests on page 11
Updated the UEFI application arguments.	See, 3.1 UEFI application arguments on page 13
Added new PAL APIs in Table 3-2: PAL APIs and UEFI interfaces.	See, 3.2 UEFI implementation of PAL APIs on page 15

Table A-12: Differences between Issue 0701-02 and Issue 0701-03

Change	Location
Updated the arguments and descriptions for UEFI Shell application and examples.	See, 3.1 UEFI application arguments on page 13

Change	Location
Updated the Linux application arguments.	See, 4.1 Linux application arguments on page 18

Table A-13: Differences between Issue 0701-03 and Issue 0701-04

Change	Location
No technical changes	-

Table A-14: Differences between Issue 0701-04 and Issue 0701-05

Change	Location
Updated the arguments and descriptions for UEFI Shell application and examples.	See, 3.1 UEFI application arguments on page 13
Added ETE module name	See, 2.2 Overview of tests on page 11
Added ETE module name and ID.	See,2.3 Test IDs on page 11.

Table A-15: Differences between Issue 0701-05 and Issue 0701-06

Change	Location
Updated the UEFI application arguments.	See, 3.1 UEFI application arguments on page 13
Added a new PAL API in Table 3-3: PAL APIs and UEFI interfaces, and ACPI tables consumed.	See, 3.2 UEFI implementation of PAL APIs on page 15

Table A-16: Differences between Issue 0701-06 and Issue 0701-07

Change	Location
Updated the UEFI application arguments.	See, 3.1 UEFI application arguments on page 13