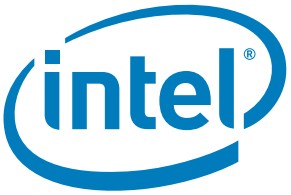


Elkhart Lake for IoT Platforms - Intel® Converged Security Engine 15.40

Compliance and Testing Guide

Revision 0.6

March 2020

Intel Confidential

Document Number: 617151

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel® products described herein. You agree to grant Intel® a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.
Intel® technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

Intel® technologies may require enabled hardware, specific software, or services activation. Check with your system manufacturer or retailer.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

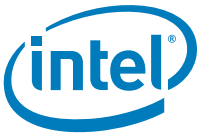
Intel® disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

All information provided here is subject to change without notice. Contact your Intel® representative to obtain the latest Intel® product specifications and roadmaps.

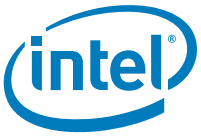
Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or visit www.intel.com/design/literature.htm.

Intel® and the Intel® logo are trademarks of Intel® Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.



Contents



Revision History

Document Number	Revision Number	Description	Revision Date
617151	0.6	• **Initial Release**	March 2020

§ §

1 Introduction

1.1 Purpose and Scope

Intel® CSE Compliance Guide for Elkhart Lake platforms is designed to provide original equipment and device manufacturers with the compliance requirements for Intel® CSE 15.40-based platforms. It provides implementation, methodology and tools to verify compliance for the various Intel® CSE and FW core components and technologies. It also provides the test environment setup information, the procedure for each test, and the expected results for the purpose of validating compliance.

Requirements contained in this document target the system BIOS, Intel® CSE and other aspects of overall platform implementation.

Note: All tests can be run without using Intel® APS device except for the Power Management tests. Without an Intel® APS device, configure Intel® PETS console SUT power settings (under the Intel® APS tab) to be Manual rather than controlled by Intel® APS device. Doing so, Intel® PETS will rely on the user to verify if the system is in S0, ACDC Power supplied, so on via user prompt questions. When doing the power management tests, it is best to do them with APS device for reliable signal measurements and results. The APS device and software will qualify if the platform transitions into the respective power state appropriately based on the signal level voltages of the board. Refer to the APS connection guide for more details on each power state signal status expectations. The board power state can be manually measured as each compliance test mentions "verify" or "observe" using a multi-meter.

1.2 Acronyms and Definitions

1.2.1 General

Acronym or Terminology	Definition
DnX	Download and Execute
FDV	Flash Descriptor Valid
FPF	Field Programmable Fuses
fTPM	Firmware Trusted Platform Module- Intel® implementation of TCG TPM 2.0 in firmware.
Intel® APS	Intel® Automatic Power Switch (Intel® APS) System State Test Device
Intel® FIT	Intel® Flash Image Tool
Intel® FPT	Intel® Flash Programming Tool
Intel® CSE	Intel® Converged Security Engine
Intel® PETS	Intel® Platform Enablement Test Suite
ISI	Intel®Safety Island
EHL	Elkhart Lake Platform
PM	Power Management
PTT	Platform Trust Technology, also known as fTPM
SPI	Serial Peripheral Interface
SUT	System Under Test

Acronym or Terminology	Definition
TCG	Trusted Computing Group
TPM	Trusted Platform Module
UFS	Universal Flash Storage

1.2.2 System States and Power Management

Acronym or Terminology	Definition
S0	A system state where power is applied to all Hardware devices and system is running normally (refer to latest industry ACPI specification).
S0i3, S3	"Sleep", Used when the user is not actively using the device. CPU in C6 (retained in shared SRAM). Sleep mode, always connected Able to wake from user or platform Screen off
S4	Longest wake latency sleeping state. OS context in Memory is saved into disk. All devices are powered off. Hibernate mode
S5	"Soft off". All devices are powered off. System Shutdown
M0	Intel® CSE FW power state where all HW power planes are activated and the host power state is S0.
M-Off	An Intel® CSE FW power state where no power is applied to the CSE subsystem. (Intel® CSE FW is shut down).
OS Hibernate	When the OS saves state information to the hard disk
Standby	When the OS state is saved to memory and resumed from the memory when mouse, keyboard, or other activity occurs that is configured as a wake event.
Shut Down	A state where the system power is off and the power cord is still connected.
Warm/Cold reset	The platform shall support restart (warm / cold) from System active state (S0) by closing the applications, initiate OS reboot sequence to bring the platform to S0 active state
Global reset	A full platform reset that includes the CSE sub system and host sub system

1.3 Intel® Platform Enablement Test Suite (Intel® PETS) Testing Guidelines

Intel recommends that customers run Intel® PETS tests whenever there are any changes in:

- BIOS
- Intel® CSE Firmware
- EC Firmware
- Board/Silicon stepping changes

The following tests should be executed in the specified order:

1. Run Intel® PETS Setup Environment Test
2. Run Integrated Clock Control test package

3. Run SPI test package
4. Run PCH Softs-trap test package
5. Run BIOS test package
6. Run Power Test package
7. Run Intel Boot Guard Test package
8. Run Signing and Manifest Test package
9. Run Manufacturing Mode Compliance Test package
10. Run PTT Test package
11. Run DnX Test package
12. Run Content Protection Test Package
13. Run FW Update Test package
14. Run Feature tests depending on the SKU

Note: PETS supports only Windows* OS on the Elkhart Lake platform in the first phase, and PETS support for Linux* and Android* will be added in the later project phases.

1.4 Boot Guard—Discrete TPM and Intel® Platform Trust Technology (Intel® PTT)

The following table shows the configuration information for the Boot Guard—Discrete TPM and Intel® Platform Trust Technology (Intel® PTT) with respect to how they work with different operating systems and firmware combinations. Refer to Boot Guard and Intel® PTT chapter for actual compliance tests.

Definitions:

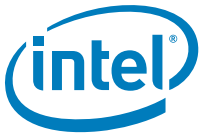
- Supported—Intel will validate this combination
- Not Supported—Intel will not validate this combination
- N/A—Not a valid combination from a validation standpoint

Table 1-1. Boot Guard—Discrete TPM and Intel® Platform Trust Technology (Intel® PTT)

Platform[1]	Intel® Active Client Manager (Intel® ACM)	Intel® CSE Firmware	Intel® PTT	TPM 2.0
Elkhart Lake Based	Intel® ACM 1.X.Y	Intel® CSE 15.40	Yes	Yes

Note:
[1]Refer to platform dashboard for POR configurations.

§ §



1.5 Elkhart Lake - Supported Operating Systems

The following table shows the different operating systems that the Elkhart Lake platform supports:

Table 1-2. Elkhart Lake - Supported Operating Systems

Feature	Windows* 10 IoT Enterprise 64-Bit	Yocto Project*-based OS 64-bit	Android* P&Q (64-bit)
Intel® Dynamic Application Loader (Intel® DAL)	YES	YES	NO
Intel® Platform Trust Technology (Intel® PTT)	YES	YES	NO
Intel® Device Protection Technology with Boot Guard	YES	YES	YES
Content Protection	Widevine* PlayReady*	Partially	Widevine*
DnX – Download and Execute	YES	YES	YES
Intel® Integrated Clock Controller Service (Intel® ICCS	YES	YES	YES
Firmware Update Tool: SPI Only Capsule: SPI	YES	YES	YES

< This page is intentionally left blank >

2 Elkhart Lake - Intel® CSE BIOS Compliance

Intel® CSE BIOS Compliance section serves as a checklist for the environment setup for the host BIOS and Intel® CSE interface testing and validation.

2.1 Intel® BIOS Compliance Test Coverage Summary

Platform, Operating System Support, How? Column describes the test methodology.

OS Support: W = Microsoft* Windows*

How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M = Manual.

Test ID	Test Case Title	PETS Package Name[1]	OS Supported	Platform	How?
BIOS_01	End of POST	Compliance_BIOS.xml	W	Elkhart Lake	A
BIOS_02	DRAM Init Done	Compliance_BIOS.xml	W	Elkhart Lake	A

Note:
[1]Tests BIOS_01, BIOS_02 and BIOS_05 can be done in one iteration after reset (no need for 3 separate resets). Tester may remove the resets from PETS SW between the tests.

2.2 End of POST

Test ID:	BIOS_01
Test Case Title:	End of POST
Objective:	Verify that the BIOS sends the END_OF_POST message when the platform is transitioning from S5 and before the BIOS boot process is done and the OS starts.
Platform:	Elkhart Lake
Mandatory/Optional:	Mandatory
Test Pass Criteria:	Test passes if the BIOS Mode displays a status of POST Boot.
Description:	At the end of BIOS POST, system BIOS must send an "END OF POST" HECI message to CSE declaring end of POST and start of OS load. BIOS must also wait for an "END of POST" response message from CSE before proceeding.
Procedure:	Verify if END_OF_POST message was sent by BIOS: • EFI Procedure: — Boot to EFI Shell — Run: "MEInfo.efi" — Check that the value of BIOS boot state is "Post boot" • Windows* Procedure: — Boot to Windows* — From elevated CMD run: "MEInfo.exe" — Check that the value of BIOS boot state is "Post boot"

2.3 DRAM INIT DONE

Test ID:	BIOS_02
Test Case Title:	DRAM Init Done
Objective:	Verify that the BIOS set the DRAM INIT Done bit
Platform:	Elkhart Lake
Mandatory/Optional:	Mandatory
Test Pass Criteria:	The BIOS is required to send the DRAM INIT Done message to Intel® CSE to indicate that BIOS has initialized DRAM memory.
Description:	This message is sent by the IA FW to indicate to Intel® CSE firmware that DRAM initialization was completed and CSE UMA is ready for use.
Procedure:	EFI Procedure: • Boot to EFI shell • Run the below command to read FWSTS#1 register • Run "PCI 0 22 0" command (add –I for verbose information) Check for offset 40h • ME Current Operating State Bit [8:6] should set to "001" - "M0 with UMA" **ME Current Operation State**: This field describes the state that CSE is currently functioning in at this moment. It's the combination of CSE power state and UMA. The "Current Operation State" is set only upon entering the true hardware state, e.g. We set it to M0 only after PLL's are locked to M0 freq and controller is set to SD. 000 – Preboot (Default) 001 – M0 with UMA 010 – M0 Power Gated 011 – Reserved 100 – M3 without UMA 101 – M0 without UMA 110 – Bring up 111 – M0 without UMA but with error

§ §

< This page is intentionally left blank >

3 Intel® Converged Security Engine (Intel® CSE) Manufacturing Mode Compliance

The Intel® Converged Security Engine Manufacturing Mode compliance chapter serves as a checklist for the environment setup for the host BIOS and Intel® CSE interface testing and validation when the Intel® CSE is in Manufacturing Mode.

The tests in this section verify that certain BIOS operations are *not* performed when the Intel® CSE is in Manufacturing Mode.

Test Environment for Intel® CSE BIOS Compliance section:

The system under test is to be configured with the Intel® CSE in manufacturing mode and Deep S4/S5 disabled.

Tools for Testing:

- Intel® Platform Enablement Test Suite—Latest version of the tool from the Intel® CSE Compliance kit release. Refer to the *Intel® Platform Enablement Test Suite User Guide* available in the Intel Compliance kit for exact instructions on how to load and setup the Intel® Platform Enablement Test Suite software.
- Compliance_MeBios_ManufacturingMode.xml—Package should be loaded to Intel Platform Enablement Test Suite in order to complete this section.

3.1 Intel® Manufacturing Mode Compliance Test Coverage Summary

OS Support', and 'How?' Columns describes the test methodology.

OS Support: W = Microsoft* Windows*, A = Android OS, L = Linux

How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M = Manual.

Form Factor: D = Desktop, M = Mobile, A = All in one, W = Workstation

Network Factor: LAN = systems with LAN interface and test is performed using LAN interface, WLAN = systems with WLAN interface and test is performed using the WLAN interface.

Test ID	Test Case Title	How?	Intel® PETS Package Name	OS Support	Form Factor	Network Factor
BIOS_04	CF9GR locking/ unlocking - Manufacturing Mode	A	Compliance_MeBios_ManufacturingMode.xml	W A L	M	LAN+WLAN; WLAN only

3.2 CF9GR Locking/Unlocking

Test ID:	**BIOS_04**
Test Case Title:	CF9GR locking/unlocking - Manufacturing Mode
Mandatory/Optional:	Mandatory
Description:	When the system is in the Intel® CSE manufacturing mode, BIOS must set the CF9GR register (Memory-mapped address at PWRMBASE register offset 1048h [bit 20]) to '0' to allow host only resets before handing control to the OS. For the Intel® FPT tool to perform a global reset with parameter/GRESET, the BIOS must keep the CF9GR setting unlocked (by setting PWRMBASE register offset 1048h [bit 31] of the same register to '0').
Objective:	For security reasons, the BIOS must ensure that CF9GR is cleared and locked before handing control to the OS in the shipping machine. But for the usage of Intel® FPT tool with /GRESET parameter in the manufacturing environment, the BIOS must ensure that CF9GR reset mode can be changed by the Intel® FPT tool. ***Note:*** The recommended allocation of PWRMBASE is 0xFE000000 in PCH BIOS Specification.
Procedure:	1. Boot the system under test to OS. 2. Intel Platform Enablement Test Suite will perform the following: a. Manually read the PCI address space at PCH B0:D22:F0 register offset 40h [bit 4] to verify the Intel® CSE Manufacturing Mode bit is equal to '1'. b. Manually read the memory-mapped address at PWRMBASE register offset 1048h [bit 20] to verify the bit is set to '0'. c. Manually read the memory-mapped address at PWRMBASE register offset 1048h [bit 31] to verify the bit is set to '0.
Test Pass/Fail Criteria:	Test passes if the PWRMBASE register offset 1048h [bit 20] = '0' and [bit 31] of the same register is '0' when the system is in the Intel® CSE manufacturing mode.

§ §

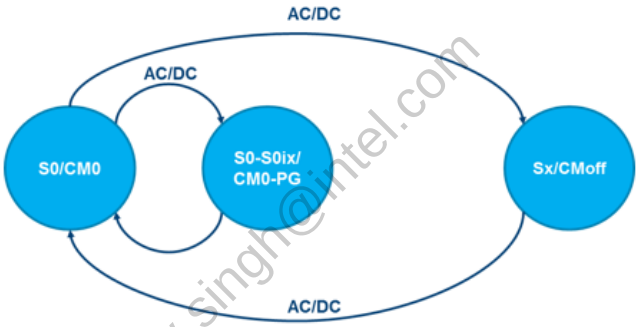
4 Intel® CSE 15.40 FW - Power Management and Stress Testing

This chapter covers system power flow transitions which involve the Intel® CSE Firmware. There are also tests in this chapter that are specifically intended to cover topics related to stress testing of the System-under-Test (SUT).

4.1 Introduction

There are certain aspects in Elkhart Lake that may differ from other platforms.

- Deep-Sx is supported by Elkhart Lake.



4.2 Test Environment Setup

- System-under-Test (SUT) can be configured in either manual configuration mode or using enterprise provisioning mode.
- Install all platform drivers (Chipset, Graphics, WLAN).
- If there is a Global reset test to pass, then the system under test should be in manufacturing mode.

4.3 Test Step Execution and Verification

The tests described in this chapter contain test steps which are executed by Intel® PETS. While Intel® PETS brings a certain level of convenience and speed to the testing process, there are times where manual verification of steps in critical towards issue triage and debug. Review the Test Step Execution and Verification Section found for Intel® CSE Power Management tests before starting any stress tests in the chapter.

Stress tests in this chapter are designed to be run individually through a large number of iterations. Some of them require changing the system configuration before being run. When performing very large numbers of iterations, the tests may each take many hours, and is some cases, several days.

Intel validation runs each of these tests the number of iterations indicated. Each OEM should decide on the tolerance level required for their boards, and choose an appropriate number of iterations.

Stress tests in this chapter are not designed to be run automatically one after the other; the test operator must place the SUT into an appropriate starting state, and then run the test cycle. However, each test individually ends with the SUT in the same state as when it started, allowing for easy iteration.

When running long iterations, ensure that the management console is set not to go to sleep, as this will pause the test.

Ensure that the SUT can boot to designated Host OS without prompting the test operator for any actions (such as scanning drivers and so forth); as this will effect stress tests which boot the SUT to the Host OS.

4.4 Tools for Testing

The following tools, as provided by Intel, may be used to execute automated tests listed herein:

- **Intel® Platform Enablement Test Suite** - Latest version of the tool is available in the Intel® CSE compliance kit release. Refer to the Intel® Platform Enablement Test Suite User Guide available in the Intel® Compliance kit for exact instructions on how to load and setup the Intel® Platform Enablement Test Suite software.
- **Intel® Automated Power Switch** – The SUT should be connected to an Intel® APS 3 unit. In case Intel® APS 3 is not available, select the Manual configuration in the Intel® PETS SUT profile configuration menu.
- **Intel® PETS Local Agent**: The agent must be installed on the SUT.
- It is recommended that power management tests (ME_PM) in this chapter be run on no less than 30% battery charge.
- It is recommended that stress tests (ME_PMST) in this chapter be run on no less than 90% battery charge.

4.5 Power Management Compliance Test Coverage Summary

Platform, Operating System Support, How? Column describes the test methodology.

OS Support: W = Microsoft* Windows *, L = Linux*, A =Android*

How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M = Manual.

Test ID	Test Case Title	PETS Package Name	OS Supported	How?
ME_PM_1	S0 to S0ix	ME_Compliance_PM.xml	W	A
ME_PM_2	S0ix to S0	ME_Compliance_PM.xml	W	A
ME_PM_3	S0 to S5 to S0	ME_Compliance_PM.xml	W	A
ME_PM_4	S5 to S0	ME_Compliance_PM.xml	W	A
ME_PM_5	Cold Reset	ME_Compliance_RST.xml	W	A

Test ID	Test Case Title	PETS Package Name	OS Supported	How?
ME_PM_6	Global Reset	ME_Compliance_RST.xml	W	A
ME_PM_7	Warm Reset	ME_Compliance_RST.xml	W	A
ME_PM_8	S0 to G3	ME_Compliance_PM.xml	W	A
ME_PM_9	S0 to S4 to S0	ME_Compliance_PM.xml	W	A
ME_PM_10	S0 to S3	ME_Compliance_PM.xml	W	A
ME_PM_11	S3 to S0	ME_Compliance_PM.xml	W	A
ME_PMST_1	Host Reset from S0	Compliance_Power_Stress.xml	W	A
ME_PMST_2	S0 to S5 to S0 via Power Button Override	Compliance_Power_Stress.xml	W	A
ME_PMST_3	S0 to S0ix to S0 via Power Button Press	Compliance_Power_Stress.xml	W	A
ME_PMST_4	S0 to S5 to S0 by means of Shutdown and Power Button Press	Compliance_Power_Stress.xml	W	A

4.6 ME_PM_1: S0 to S0ix

Test ID:	ME_PM_1.1
Test Case Title:	S0 to S0ix by means of Power Button Suspend (DC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0ix
Test Pass Criteria:	The test passes if the SUT moves to S0ix and the Intel® CSE is working properly with no flash logs found
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify that a DC Battery is connected to the SUT, and that it is charged 4. Set the SUT power source to DC Only. 5. Verify CSE is working properly. 6. Suspend the SUT via the Power Button 7. Verify the SUT is in S0ix. 8. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

Test ID:	ME_PM_1.2
Test Case Title:	S0 to S0ix by means of Power Button Suspend (AC+DC, AConly)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0ix
Test Pass Criteria:	The test passes if the SUT moves to S0ix and the Intel® CSE is working properly with no flash logs found.

Test ID:	ME_PM_1.2
Procedure:	1. Set the SUT power source to AC+DC, otherwise AC only 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly. 4. Suspend the SUT via the Power Button 5. Verify the SUT is in S0ix. 6. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

4.7 ME_PM_2: S0ix to S0

Test ID:	ME_PM_2.1
Test Case Title:	S0ix to S0 by means of Power Button Press (DC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0ix to S0
Test Pass Criteria:	The test passes if the SUT moves to S0 and the Intel® CSE is working properly with no flash logs found.
Procedure:	1. Set the SUT power source to AC+DC only. 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager (This is applicable only in Windows) 4. Verify that DC battery is connected to the SUT, and that it is charged 5. Set the SUT power source to DC-only. 6. Move the SUT to S0Ix via the Power Button 7. Verify the SUT is in S0ix 8. Briefly press the Power Button on the SUT 9. Verify that the SUT is in S0 10. Verify that the Host OS on the SUT is available. 11. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**0**xxxxxx b. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. (This is applicable in Windows only) 12. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

Test ID:	ME_PM_2.2
Test Case Title:	S0ix to S0 by means of Power Button Press (AC+DC, AC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0ix to S0
Test Pass Criteria:	The test passes if the SUT moves to S0 and the Intel® CSE is working properly with no flash logs found

Test ID:	ME_PM_2.2
Procedure:	1. Set the SUT power source to AC+DC. 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager (This is applicable only in Windows) 4. Move the SUT to S0ix via the Power Button 5. Verify the SUT is in S0ix 6. Briefly press the Power Button on the SUT 7. Verify that the SUT is in S0 8. Verify that the Host OS on the SUT is available. 9. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**0**xxxxxx b. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. (This is applicable Windows only) 10. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

4.8 ME_PM_3: S0 to S5 to S0

Test ID:	ME_PM_3.1
Test Case Title:	S0 to S5 to S0 by means of Host OS Shutdown (DC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S5 to S0
Test Pass Criteria:	The test passes if the SUT moves to S5 to S0 and the Intel® CSE is working properly with no flash logs found.
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly. 4. Verify that a DC Battery is connected to the SUT, and that it is charged 5. Set the SUT power source to DC only 6. Shutdown the SUT via the Host OS 7. Verify the SUT is in S5. 8. Press the Power Button on the SUT 9. Verify the SUT is in S0. 10. Verify that the Host OS on the SUT is available. 11. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**0**xxxxxx 12. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. (This is applicable in Windows Only) 13. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

Test ID:	ME_PM_3.2
Test Case Title:	S0 to S5 to S0 by means of Host OS Shutdown (AC+DC, AC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S5 to S0
Test Pass Criteria:	The test passes if the SUT moves to S5 to S0 and the Intel® CSE is working properly with no flash logs found

Test ID:	ME_PM_3.2
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly. 4. Shutdown the SUT via the Host OS 5. Verify the SUT is in S5. 6. Press the Power Button on the SUT at least 7. Verify the SUT is in S0. 8. Verify that the Host OS on the SUT is available. 9. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**0**xxxxxx 10. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. (This is applicable only in Windows) 11. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

4.9 ME_PM_4: S5 to S0

Test ID:	ME_PM_4.1
Test Case Title:	S5 to S0 by means of Power Button Press (DC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S5 to S0
Test Pass Criteria:	The test passes if the SUT moves to S0 and the Intel® CSE is working properly with no flash logs found
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager (This is only applicable in Windows) 4. Verify that a DC Battery is connected to the SUT, and that it is charged 5. Set the SUT power source to DC Only 6. Shutdown the SUT via the Host OS 7. Verify the SUT is in S5. 8. Briefly press the Power Button on the SUT 9. Verify the SUT is in S0. 10. Verify that the Host OS on the SUT is available. 11. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**0**xxxxxx b. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. (This is only applicable in Windows) 12. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

Test ID:	ME_PM_4.1
Test Case Title:	S5 to S0 by means of Power Button Press (DC only)

Test ID:	ME_PM_4.1
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S5 to S0
Test Pass Criteria:	The test passes if the SUT moves to S0 and the Intel® CSE is working properly with no flash logs found
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager (This is only applicable in Windows) 4. Verify that a DC Battery is connected to the SUT, and that it is charged 5. Set the SUT power source to DC Only 6. Shutdown the SUT via the Host OS 7. Verify the SUT is in S5. 8. Briefly press the Power Button on the SUT 9. Verify the SUT is in S0. 10. Verify that the Host OS on the SUT is available. 11. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**0**xxxxxx b. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. (This is only applicable in Windows) 12. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

4.10 ME_PM_5: Cold Reset

Test ID:	ME_PM_5.1
Test Case Title:	S0 to S0 by means of Cf9 Cold Reset (AC+DC, AC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0
Test Pass Criteria:	The test passes if the SUT resets to S0 and the Intel® CSE is working properly with no flash logs found
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager (This is only applicable in Windows only) 4. Record the Host OS last boot time on the SUT (to verify reset execution) 5. Ensure the Cf9h Global Reset is cleared to 0b 6. Perform a cold reset to the SUT by writing Eh to IO register CF9h. PCI configuration space for CF9h (Bus:Device:Function) is (0,F,0) and the offset is 0x40 7. Verify the SUT is in S0. 8. Verify the Host OS last boot time on the SUT does not match the boot time recorded before reset or that HOST OS is unavailable. 9. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. (This is only applicable in Windows only) 10. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

4.11 ME_PM_6: Global Reset

Test ID:	ME_PM_6.1
Test Case Title:	S0 to S0 by means of Cf9 Global Reset (AC+DC, AC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0
Test Pass Criteria:	The test passes if the SUT resets to S0 and the Intel® CSE is working properly with no flash logs found
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager (This is only applicable in Windows) 4. Record the Host OS last boot time on the SUT (to verify reset execution) 5. Send CBM_Promote_WARM_RST_TO_GRST message 6. Perform a warm reset to the SUT by writing 6h or Eh to IO register CF9h. PCI configuration space for CF9h (Bus:Device:Function) is (0,F,0) and the offset is 0x40. ***Note:*** Perform a warm reset after sending HECI command mentioned in test step 5 will perform Global Reset. 7. Verify the SUT is in S0. 8. Verify the Host OS last boot time on the SUT does not match the boot time recorded before reset or that HOST OS is unavailable. 9. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**2**xxxxxx or 0x6**B**xxxxxx b. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager (This is applicable in Windows only) 10. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

Test ID:	ME_PM_6.2
Test Case Title:	S0 to S0 by means of Cf9 Global Reset (DC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0
Test Pass Criteria:	The test passes if the SUT resets to S0 and the Intel® CSE is working properly with no flash logs found

Test ID:	ME_PM_6.2
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager (This is only applicable in Windows) 4. Verify that the DC Battery is connected, and that it is charged. 5. Set the SUT power source to DC-only. 6. Record the Host OS last boot time on the SUT (to verify reset execution) 7. Send CBM_Promote_WARM_RST_TO_GRST message 8. Perform a warm reset to the SUT by writing 6h or Eh to IO register CF9h.PCI configuration space for CF9h (Bus:Device:Function) is (0,F,0) and the offset is 0x40. Note: Perform a warm reset after sending HECI command mentioned in test step 7 will perform Global Reset. 9. Verify the SUT is in S0. 10. Verify the Host OS last boot time on the SUT does not match the boot time recorded before reset or that HOST OS is unavailable. 11. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**2**xxxxxx or 0x6**B**xxxxxx b. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager (This is applicable in Windows only) 12. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

4.12 ME_PM_7: Warm Reset

Test ID:	ME_PM_7.1
Test Case Title:	S0 to S0 by means of Host OS Restart (AC+DC, AC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0
Test Pass Criteria:	The test passes if the SUT resets to S0 and the Intel® CSE is working properly with no flash logs found
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager (This is applicable in Windows only) 4. Record the Host OS last boot time on the SUT (to verify the reset execution). 5. Perform a warm reset of the SUT via Host OS graceful restart. 6. Verify the SUT is in S0. 7. Verify the Host OS last boot time on the SUT does not match the boot time recorded before reset, or that the Host OS is unavailable. 8. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**9**xxxxxx b. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. (This is applicable in Windows only) 9. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

Test ID:	**ME_PM_7.4**
Test Case Title:	S0 to S0 by means of Host OS Restart (DC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0
Test Pass Criteria:	The test passes if the SUT resets to S0 and the Intel® CSE is working properly with no flash logs found
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager (This is applicable in Windows only) 4. Record the Host OS last boot time on the SUT (to verify the reset execution). 5. Verify that DC battery is connected, and that it is charged 6. Connect the SUT power source to DC-only 7. Perform a warm reset of the SUT via Host OS graceful restart. 8. Verify the SUT is in S0. 9. Verify the Host OS last boot time on the SUT does not match the boot time recorded before reset, or that the Host OS is unavailable. 10. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**9**xxxxxx b. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. (This is applicable in Windows only) 11. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

Test ID:	**ME_PM_7.5**
Test Case Title:	S0 to S0 by means of CF9 Warm Reset (AC+DC, AC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0
Test Pass Criteria:	The test passes if the SUT resets to S0 and the Intel® CSE is working properly with no flash logs found
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager (This is only applicable in Windows) 4. Record the Host OS last boot time on the SUT (to verify the reset execution). 5. Perform a warm reset of the SUT by writing 6h to IO register CF9h. 6. Verify the SUT is in S0. 7. Verify the Host OS last boot time on the SUT does not match the boot time recorded before reset, or that the Host OS is unavailable. 8. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x69xxxxxx b. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. (This is only applicable in Windows) 9. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

4.13 ME_PM_8: S0 to G3

Test ID:	ME_PM_8.1
Test Case Title:	S0 to G3 by means of Power Loss (AC+DC, AC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0
Test Pass Criteria:	The test passes if the SUT moves from S0 to G3 and the Intel® CSE is working properly with no flash logs found
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly 4. Remove power from the SUT via AC-detach, and if necessary also via DC-detach. Wait for 10 seconds before continuing to allow full power drain from the SUT. 5. Verify the SUT is in G3. 6. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

4.14 ME_PM_9: S0 to S4 to S0

Test ID:	ME_PM_9.1
Test Case Title:	S0 to S4 to S0 by means of Host OS Hibernate (DC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S4
Test Pass Criteria:	The test passes if the SUT moves to S5 and the Intel® CSE is working properly with no flash log found.
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly. 4. Verify that a DC Battery is connected to the SUT, and that it is charged 5. Set the SUT power source to DC only 6. Hibernate the SUT via the Host OS 7. Verify the SUT is in S4. 8. Press the Power Button on the SUT 9. Verify the SUT is in S0. 10. Verify that the Host OS on the SUT is available. 11. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**0**xxxxxx 12. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. (This is applicable in Windows only) 13. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

Test ID:	ME_PM_9.2
Test Case Title:	S0 to S4 to S0 by means of Host OS Hibernate (AC+DC, AC only)
Mandatory/Optional:	Mandatory

Test ID:	ME_PM_9.2
Objective:	This test checks the SUT power flow from S0 to S4
Test Pass Criteria:	The test passes if the SUT moves to S4 and the Intel® CSE is working properly with no flash log found.
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly. 4. Hibernate the SUT via the Host OS 5. Verify the SUT is in S4. 6. Press the Power Button on the SUT 7. Verify the SUT is in S0. 8. Verify that the Host OS on the SUT is available. 9. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**0**xxxxxx 10. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. (This is applicable in Windows only) 11. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

4.15 ME_PM_10: S0 to S3

Test ID:	ME_PM_10.1
Test Case Title:	S0 to S3 (DC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S3
Test Pass Criteria:	The test passes if the SUT moves to S0ix and the Intel® CSE is working properly with no flash logs found
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify that a DC Battery is connected to the SUT, and that it is charged 4. Set the SUT power source to DC Only. 5. Verify CSE is working properly. 6. Suspend the SUT via the Host OS 7. Verify the SUT is in S3. 8. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

Test ID:	ME_PM_10.2
Test Case Title:	S0 to S3 (AC+DC, AConly)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S3
Test Pass Criteria:	The test passes if the SUT moves to S0ix and the Intel® CSE is working properly with no flash logs found.

Test ID:	ME_PM_10.2
Procedure:	1. Set the SUT power source to AC+DC, otherwise AC only 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly. 4. Suspend the SUT via the Host OS 5. Verify the SUT is in S3. 6. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

4.16 ME_PM_11: S3 to S0

Test ID:	ME_PM_11.1
Test Case Title:	S3 to S0 (DC only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S3 to S0
Test Pass Criteria:	The test passes if the SUT moves to S0 and the Intel® CSE is working properly with no flash logs found
Procedure:	1. Set the SUT power source to AC+DC 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager (This is only applicable in Windows) 4. Verify that a DC Battery is connected to the SUT, and that it is charged 5. Set the SUT power source to DC Only 6. Suspend the SUT via the Host OS 7. Verify the SUT is in S3. 8. Briefly press the Power Button on the SUT 9. Verify the SUT is in S0. 10. Verify that the Host OS on the SUT is available. 11. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**0**xxxxxx b. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. (This is only applicable in Windows) 12. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

Test ID:	ME_PM_11.2
Test Case Title:	S3 to S0 (AC+DC, AConly)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S3 to S0
Test Pass Criteria:	The test passes if the SUT moves to S0 and the Intel® CSE is working properly with no flash logs found

Test ID:	ME_PM_11.2
Procedure:	1. Set the SUT power source to AC+DC where supported; otherwise AC-only 2. Bring the SUT to base state of S0 and confirm that the host OS is available 3. Verify CSE is working properly a. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager (This is only applicable in Windows) 4. Suspend the SUT via the Host OS 5. Verify the SUT is in S3. 6. Briefly press the Power Button on the SUT 7. Verify the SUT is in S0. 8. Verify that the Host OS on the SUT is available. 9. Verify CSE is working properly a. Verify that the second nibble of the FWSTS 2 register on the SUT have a value of 0x6**0**xxxxxx b. Ensure that yellow bang is not seen on the MEI and SPD drivers in the device manager. (This is only applicable in Windows) 10. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS")

4.17 ME_PMST_1: Host Reset from S0

Test ID:	ME_PMST_1
Test Case Title:	Host Reset from S0 (DC Only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0 by means of Host Reset
Test Pass Criteria:	The test passes if the all steps are completed successfully for at least the recommended number of iterations as set by the OEM per the tolerance level of the system design with no flash logs found. Suggested Iterations: >=750
Procedure:	1. Set the SUT power source to DC 2. Bring the SUT to base state of S0 and confirm the Host OS is available. 3. Verify CSE is working properly 4. Record the Host OS last boot time on the SUT (to verify reset execution) 5. Perform a warm reset of the SUT by performing a host reset. 6. Verify the SUT is in S0. 7. Verify that Host OS on the SUT is available. 8. Verify CSE is working properly 9. Verify the Host OS last boot time on the SUT does not match the boot time recorded before reset. 10. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS") Repeat the procedure for the remaining number of cycles desired in the stress test.

4.18 ME_PMST_2: S0 to S5 to S0 via Power Button Override

Test ID:	ME_PMST_2
Test Case Title:	S0 to S5 to S0 by means of Power Button override cycle (DC Only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S5 to S0 by means of Power Button Override
Test Pass Criteria:	The test passes if the all steps are completed successfully for at least the recommended number of iterations as set by the OEM per the tolerance level of the system design. with no flash logs found Suggested Iterations: >=750
Procedure:	1. Set the SUT power source to DC 2. Bring the SUT to base state of S0 and confirm the Host OS is available 3. Verify CSE is working properly 4. Shutdown the SUT via Power Button Press . 5. Verify that the SUT is in S5 6. Briefly press the Power Button on the SUT. 7. Verify the SUT is in S0. 8. Verify that Host OS on the SUT is available. 9. Verify CSE is working properly 11. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS") Repeat the procedure for the remaining number of cycles desired in the stress test.

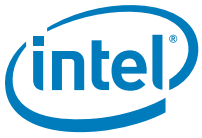
4.19 ME_PMST_3: S0 to S0ix to S0 via Power Button Press

Test ID:	ME_PMST_3
Test Case Title:	S0 to S0ix to S0 by means of Suspend and Power Button Press (DC Only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S0Ix to S0 by means of Suspend and Power Button Press
Test Pass Criteria:	The test passes if the all steps are completed successfully for at least the recommended number of iterations as set by the OEM per the tolerance level of the system design with no flash logs found. Suggested Iterations: >=750
Procedure:	1. Set the SUT power source to DC 2. Bring the SUT to base state of S0 and confirm the Host OS is available 3. Verify CSE is working properly 4. Suspend the SUT via the Power Button 5. Verify that the SUT is in S0ix 6. Briefly press the Power Button on the SUT. 7. Verify the SUT is in S0. 8. Verify that the Host OS on the SUT is available. 9. Verify CSE is working properly 10. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS") Repeat the procedure for the remaining number of cycles desired in the stress test.

4.20 ME_PMST_4: S0 to S5 to S0 via Shutdown and Power Button Press

Test ID:	**ME_PMST_5**
Test Case Title:	S0 to S5 to S0 by means of Shutdown and Power Button Press (DC Only)
Mandatory/Optional:	Mandatory
Objective:	This test checks the SUT power flow from S0 to S5 to S0 by means of Shutdown and Power Button Press
Test Pass Criteria:	The test passes if the all steps are completed successfully for at least the recommended number of iterations as set by the OEM per the tolerance level of the system design with no flash logs found. Suggested Iterations: >=750
Procedure:	1. Set the SUT power source to DC 2. Bring the SUT to base state of S0 and confirm the Host OS is available 3. Verify CSE is working properly 4. Shutdown the SUT via the Host OS 5. Verify that the SUT is in S5 6. Briefly press the Power Button on the SUT. 7. Verify the SUT is in S0. 8. Verify that the Host OS on the SUT is available. 9. Verify CSE is working properly 10. Check if there are any flash log. Success if there is no flash log.(Can test flash log by "MEInfo -FWSTS") Repeat the procedure for the remaining number of cycles desired in the stress test.

§ §



< This page is intentionally left blank >

5 Serial Peripheral Interface (SPI) Configuration

The test cases in this chapter are created to verify the correct configuration of the Elkhart Lake SoC SPI Host Controller. Test cases in this section verify implementation of SPI Dual and Quad I/O Fast Read, SPI Flash Descriptor mode, and ensure compliance with Intel® CSE requirements.

5.1 Test Environment Setup

The System Under Test (SUT) is to be configured in manual configuration mode a with wired LAN or wireless LAN dynamic IP address. The DHCP server connecting the SUT and Management Console (MC) must be configured to ensure that the wired LAN and wireless LAN addresses reside on separate subnets. The MC could be a laptop or desktop system running a version of Windows* supported by PETS. The network configuration consists of a hub or switch, network cables, and a wireless Access Point (AP).

5.2 Tools for Testing

Intel® Flash Image Tool (fit.exe)

Intel® Flash Programming Tool (Intel® FPT) - is available in Windows* 32-bit (fptw.exe), Windows* 64-bit (fptw64.exe) operating systems, EFI 32-bit and EFI 64-bit.

Intel® Platform Enablement Test Suite (Intel® PETS) - Latest version of the tool is available in the Intel® CSE compliance kit release. Refer to the Intel® Platform Enablement Test Suite User Guide available in the Intel® Compliance kit for exact instructions on how to load and setup the Intel® Platform Enablement Test Suite software.

WinPE Tools: When using Windows* FW tools in WinPE, remember to load the MEI driver at every boot. This can be done by: *X:\Windows\System32>drvload.exe <path>\MEI.inf.* MEI.inf can be found in every FW kit release.

5.3 SPI Compliancy Test Coverage Summary

Platform, Operating System Support, How? Column describes the test methodology.

OS Support: W = Microsoft* Windows*

How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M = Manual.

Test ID	Test Case Title	PETS Package Name	OS Supported	How?
SPI_01	Descriptor Mode Test	Compliance_SpiFlashConfiguration.xml	W	A
SPI_02	Serial Flash Discoverable Parameter Test	Compliance_SpiFlashConfiguration.xml	W	A
SPI_03	4 Kbytes Erasable Blocks Test	Compliance_SpiFlashConfiguration.xml	W	A
SPI_04	SFDP version 1.0 test	Compliance_SpiFlashConfiguration.xml	W	A
SPI_05	SPI Flash Size Test	Compliance_SpiFlashConfiguration.xml	W	A
SPI_06	SPI Flash Vendor Specific Capabilities (VSCC) Test	Compliance_SpiFlashConfiguration.xml	W	A
SPI_07	Flash Descriptor Security Override Test	Compliance_SpiFlashConfiguration.xml	W	I

5.4 Descriptor Mode Test

Test ID:	**SPI_01**
Test Case Title:	Descriptor Mode Test
Platform:	Elkhart Lake (SPI-based boot Only)
Mandatory/Optional:	Mandatory
Objective:	Verify the SPI flash controller in the SoC is operating in Descriptor Mode.
Test Pass Criteria:	Test passes if FDV bit is set to 1b.
Description:	Descriptor Mode is required for all SKUs of the SoC to ensure proper operation of features such as the Intel® CSE and SoC softstraps.
Procedure:	1. Boot to the target OS. 2. Verify the Flash Descriptor Valid bit is 14 in the Hardware Sequencing Flash Status (HSFS) register (SPIBAR + 04h) is set to 1b.

5.5 Serial Flash Discoverable Parameter Test

Test ID:	**SPI_02**
Test Case Title:	Serial Flash Discoverable Parameter (SFDP) Test
Platform:	Elkhart Lake (SPI-based boot Only)
Mandatory/Optional:	Mandatory
Objective:	Verify that the SPI flash controller in the SoC is able to detect a valid SFDP table in the SPI flash device.
Test Pass Criteria:	Test passes if all steps return expected values.
Description:	Proper SFDP support in the SPI flash device may be used to enable advanced SPI features like the Quad IO Fast Read.
Procedure:	1. Boot to target OS. 2. Does flash device 0 in the SUT supports SFDP? • If Yes, — Verify that the Component Property Parameter Table Valid (CPPTV) bit 31 of the Vendor Specific Component Capabilities 0 register (VSCC0[4]) is set to 1b. • If No, — Inform the test operator that SFDP support in the SPI flash device may be used to enable advanced SPI features like the Quad I/O Fast Read[3]. 3. Read the number of SPI parts by means of the Number of Components (NC) bits [9:8] in the Flash Map 0 (FLMAP0) register at (FDBAR + 14h). • If the number of components is 01b (2 Components) continue to next step else end test. 4. Does flash device 1 in the SUT supports SFDP? • If Yes, — Verify that the Component Property Parameter Table Valid (CPPTV) bit 31 of the Vendor Specific Component Capabilities 1 register (VSCC1[4]) is set to 1b. • If No, — Inform the test operator that SFDP support in the SPI flash device may be used to enable advanced SPI features like that Quad I/O Fast Read[3]. ***Notes:*** 1. VSCC0 register is located at (VTBA[4] + C4h). 2. VSCC1 register is located at (VTBA[4] + C4h + (n*8)h), where n=1. 3. Test considered pass, this is just additional information to user. 4. Refer to SPI Programming Guide for details of these registers.

5.6 4 Kbytes Erasable Blocks Test

Test ID:	**SPI_03**
Test Case Title:	4 Kbytes Erasable blocks Test
Platform:	Elkhart Lake (SPI-based boot Only)
Mandatory/Optional:	Mandatory
Objective:	Verify the SPI flash device supports uniform 4 Kbytes erasable blocks.
Test Pass Criteria:	Test passes if all steps return expected values.
Description:	The SPI Flash device must provide uniform 4 Kbytes erasable blocks/sectors throughout the entire part. This is required by Intel® CSE firmware.
Procedure:	Part 1: Verify registers. 1. Boot to the target OS. 2. Verify the SUT is operating in Descriptor Mode by confirming that the Flash Descriptor Valid (FDV) bit 14 in the Hardware Sequencing Flash Status (HSFS) register (SPIBAR + 04h) has been set to '1'. 3. Verify all flash components support 4 Kbytes erasable blocks by confirming that the Block/Sector Erase Size (BERASE) bits [4:3] in the Hardware Sequencing Flash Status (HSFS) register (SPIBAR + 04) are set to 01b. Part 2: Check against SPI flash device datasheet. 1. Using the "MEInfo"[1] tool, read the SPI flash device ID from the SUT. 2. Verify the SPI flash device ID(s) read from the SUT are found in the vsccommn.bin[2] SPI part registry cached in Intel® PETS. ***Notes:*** 1. The "MEInfo" tool is part of the Intel® Converged Security Engine Firmware release package, under System Tools folder. 2. The vsccommn.bin file will be updated relative to the latest official version for each Intel® PETS release.

5.7 SFDP Version 1.0 Test

Test ID:	**SPI_04**
Test Case Title:	SFDP version 1.0 and above test
Platform:	Elkhart Lake (SPI-based boot Only)
Mandatory/Optional:	Mandatory
Objective:	Verify SPI part is Elkhart Lake SFDP requirement compliant. This is SPI requirement to have minimum SFDP of v1.0. Elkhart Lake SoC requires SPI flash devices support JEDEC standard JESD216 SDFDP v1.0 (Serial Flash Discoverable Parameters). Revision A (JESD216A - v1.5) or later is strongly recommended but not mandatory. Refer Elkhart Lake SoC SPI and SMIP programming guide for more details (RDC#: 571036)
Test Pass Criteria:	SFDP version is 1.0 or above.
Description:	Intel SoC SKUs each have different requirements for SPI flash. This test verifies that the SPI flash device used meets the minimum SFDP version (1.5) required to use for Elkhart Lake platform.
Procedure:	Refer to the diagram and the table below for more details. 1. Locate the SFDP table in the SPI part 2. Read hex byte address offset 0x04, SFDP Minor Revision [7:0] and SFDP Major Revision [15:8] 3. Ensure Major.Minor revision is 1.0 or above

Figure 5-1. SFDP Mapping Diagram 1

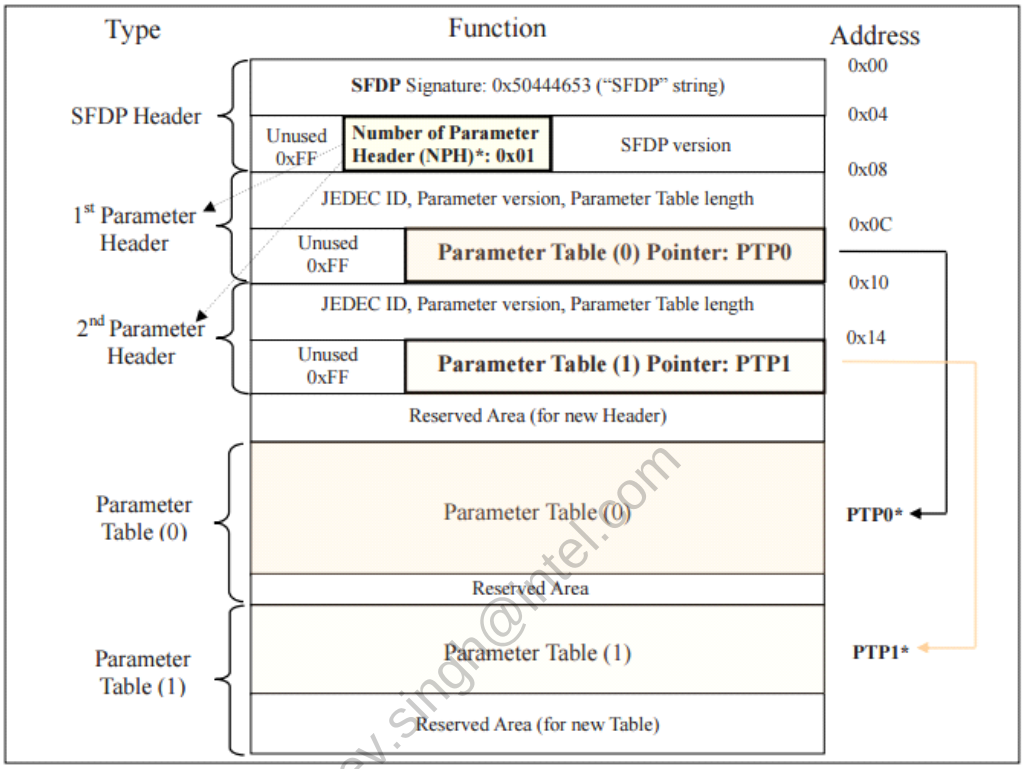


Figure 5-2. SFDP Mapping Diagram 2

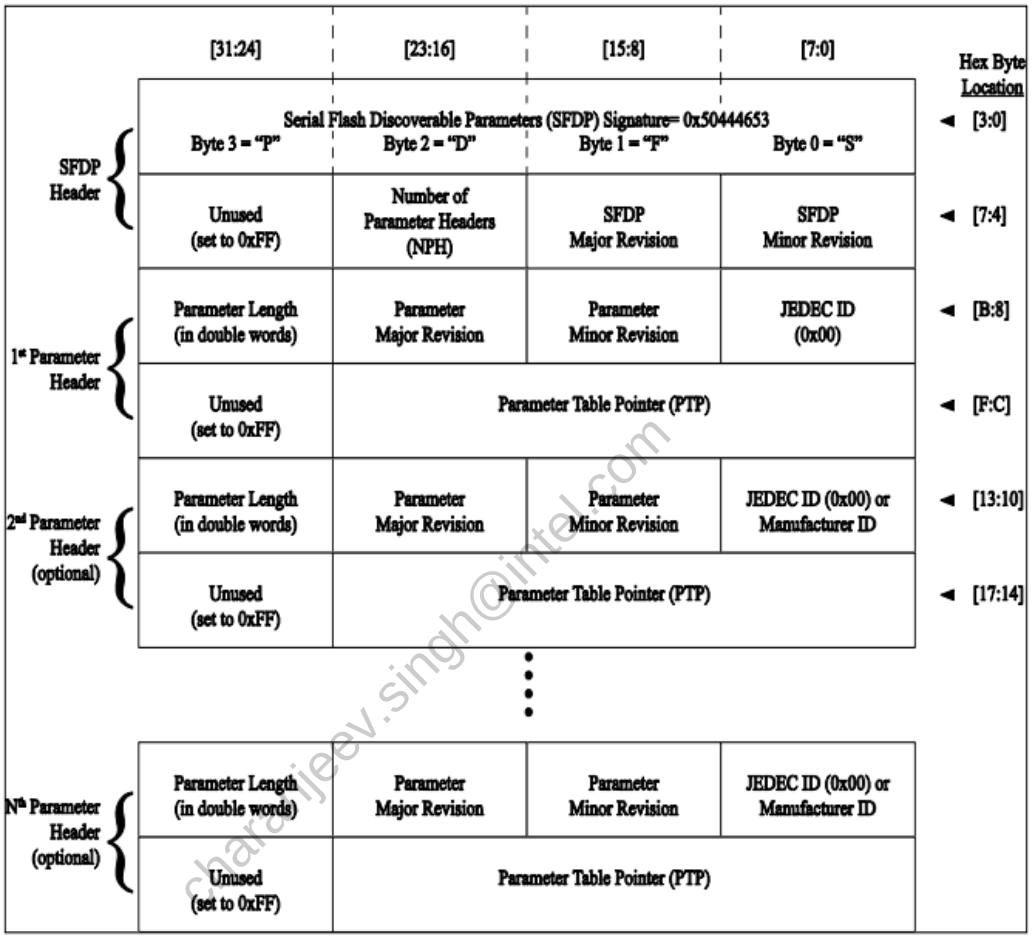
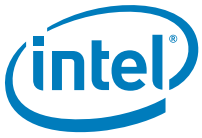


Table 5-1. SFDP Parameter Table Definition and Content

Table	Length	Maker	Definition Function
Parameter Table(0)	9 DWords	JEDEC STD	Sector Size, BP bits type
			4K Erase opcode
			Read mode, Address mode, DTR mode
			Flash density
			Read mode interface, Mode bits, dummy cycle
			Sector Size, Sector erase opcode
Parameter Table (1)	4 DWords	[SPI Vendor defines]	Voltage range
			Reset#, Hold# pin, Deep Power Down, SW Reset function, SuspendResume, Wrap-Around read
			Security function

5.8 SPI Flash Size Test

Test ID:	**SPI_05**
Test Case Title:	SPI Flash Size Test
Platform:	Elkhart Lake (SPI-based boot Only)
Mandatory/Optional:	Mandatory
Objective:	Verify the correct SPI flash size is used for a given SoC SKU contained in the SUT.
Test Pass Criteria	The test passes if the following conditions is true: 1. The flash components' sizes in the SUT are equal to the size stated in the SPI device manufacturer datasheet.
Description:	Intel SoC SKUs each have different requirements for SPI flash sizes. This test verifies that the SPI flash device has enough space to store the whole SPI image created by Intel® FIT tool.
Procedure:	1. Boot to target OS. 2. Read following information from SPI Flash Descriptor in the SUT: a. The number of SPI parts by means of the Number of Components (NC) bits [9:8] in the Flash Map 0 (FLMAP0) register at (FDBAR + 14h). b. The size of the first flash component by means of the "Flash Density" in SFDP table above in Parameter Table (0) c. If the number of components is 01b (2 Components), read the size of the second flash component by means of the "Flash Density" in SFDP table above in Parameter Table (0) 3. Compare the SUT flash size against the: a. SPI flash device manufacturer datasheet[1]. ***Note:*** 1. Intel® PETS will maintain a list of SPI flash device sizes.



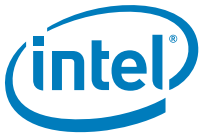
5.9 SPI Flash Vendor Specific Capabilities (VSCC) Test

Test ID:	**SPI_06**
Test Case Title:	SPI Flash Vendor Specific Component Capabilities (VSCC) Test.
Platform:	Elkhart Lake (SPI-based boot Only)
Mandatory/Optional:	Mandatory
Objective:	To verify VSCCn registers in memory mapped space and VSCC table in SPI Flash Descriptor is configured correctly.
Test Pass Criteria:	Test results pass if VSCC0 or VSCC0 and VSCC1, and the VCSS table in SPI Flash Descriptor align with the Intel® CSE VSCC and SPI flash device manufacturer datasheet settings.
Description:	The VSCC registers are defined in two places. Host-based VSCCn registers (for example, VSCC0 and VSCC1) in memory mapped space and the Intel® CSE VSCC Table in the SPI Flash Descriptor. Intel® CSE only uses the VSCC table in the SPI Flash Descriptor, while the memory map VSCCn registers are used by BIOS. The Intel® CSE VSCC table is created using the FIT tool by ODM/OEM, while the memory mapped VSCCn registers are programed by BIOS. Incorrect VSCCn registers configuration may affect SPI flash functionality and also may lead to premature flash device wear out.
Procedure:	1. Boot to the target OS. 2. Read the Vendor Specific Component Capabilities Registers (VSCCn), in the memory mapped space, where these register are located at (SPIBAR + C4h) and (SPIDBAR + C4h + (1 * 8)h) respectively. 3. Verify the VSCCn values with the SPI Flash device manufacturer datasheet. 4. Read the VSCC table from the SPI flash device on the target system. The base address of the table is located at offset (FDBAR1 + EFCh). The Intel® CSE VSCC Table Base Address (VTBA) and the Intel® CSE VSCC Table Length (VTL) are located at (FDBAR + EFCh). 5. Every record in the table is 2 DWORDs long, the first 32 bits contain the SPI flash device's JEDEC ID, and the following 32 bits represent its VSCC value. 6. Iterate through the VSCC table searching for the matching JEDEC ID of the SPI devices in use on the SUT and verify the associated VSCC values matches both the SPI flash device manufacturer datasheet and the Intel® CSE VSCC value. ***Note:*** FDBAR is located at address 0 of the SPI flash device chip select 0.

5.10 Flash Descriptor Security Override Test

Test ID:	**SPI_07**
Test Case Title:	Flash Descriptor Security Override Test
Platform:	Elkhart Lake (SPI-based boot Only)
Mandatory/Optional:	Mandatory
Objective:	This test is to verify the platform has the ability to enable and disable Intel® CSE manufacturing mode, and to be able to reprogram the entire SPI flash.
Test Pass Criteria:	Test passes if FDOPSS bit is set to '1' by default and set to '0' when intending to enter Intel® CSE Test Mode.
Description:	This boots the platform in Intel® CSE Test Mode. This gives the ability to override Flash descriptor permissions debug/repair depot environments. This must NOT be default behavior. Flash Descriptor Override (FDP) is GPIO_42. Check Elkhart Lake Platform Design Guide (PDG) document for more details (RDC# 567247).
Procedure:	Elkhart Lake RVP for FDO is mapped S2 jumper SW7D1. 1. Boot platform with **NOT** having FDO asserted high. Verify that FDOPSS is set to '1'. FDOPSS is in MMIO space (SPIBAR + 0x4) bit 13 2. Boot platform with having FDO asserted high. Verify that FDOPSS is set to '0'. FDOPSS is in MMIO space (SPIBAR + 0x4) bit 13. This assertion of FDO can be with a jumper or through another external mechanism. Care should be taken to ensure that assertion of this mechanism to assert FDO **cannot** be done remotely. PETS will help automate testing of this capability. Perform the test by enabling "State after G3 to S5" at BIOS setting. **Alternate Procedure** 1. Configure the platform with Intel® CSE Firmware. 2. Use FPT -f to flash new image. This test should fail. 3. Use the physical jumper to override the protection (asserts FDO GPIO_42 high). Boot system from G3 state 4. Use FPT -f to flash new image. This test should now pass

§ §



6 Common Services

This chapter covers Intel® CSE related features and technologies. Among those are the following features, which require BIOS and/or system integration:

- Intel® CSE Firmware Update

6.1 Test System Configuration

Each test in this chapter contains a table describing the system configuration to which the test is applicable. Below is an example environment for a given test:

Form Factor	System Power Model
☑ Desktop ☑ Workstation ☑ Mobile	☑ Standard ☑ Modern Standby or InstantGo*

Form Factor: Describes the kind of system for which the test is applicable. These tests cover feature availability for associated platform.

System Power Model: Describes under which System Power Model the test is applicable under. A system with 'Standard' configuration follows traditional OS power model in sending the system to Sleep results in a S3 or DeepS3 resting system state. Systems that support Modern Standby or Microsoft* Windows InstantGo* moves to S0 Low Power Idle state upon being sent to Sleep. This is usually defined by feature support relative to the operating system in conjunction with BIOS and system device support, but may also be due to the nature of the operating system itself relative to the goals of the test.

6.2 Test Coverage Summary

The following describes columns in the test coverage summary below. The **Test ID** is the reference identifier for the test in this document and any related tools, which reference this document. The **Title** is the name of the test. The Requirement (**Req.**) column describes the requirement for test execution. The **Form Factor**, **OS** (Operating System) indicate the applicable test system configuration (refer the Section 6.1 for details). **How?** column describes the test methodology.

Req.: M = Mandatory, C = Conditional†, and O = Optional

† Considered the same as Mandatory but with exemptions. Refer test for details.

Form Factor: D = Desktop, M = Mobile, and W = Workstation

Power Model: S = Standard, and M/I = Modern Standby or Microsoft* Windows InstantGo* (refer above for details)

How?: A = Fully automated using Intel® PETS, I = Interactive using Intel® PETS automation, and M = Manual

Table 6-2. Intel® CSE Test Coverage Summary

Test ID	Title	Req.	Form Factor D M W	Power Model S M/I	How?
Intel CSE Firmware Update					

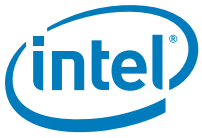


Table 6-2. Intel® CSE Test Coverage Summary

Test ID	Title	Req.	Form Factor D M W	Power Model S M/I	How?
CS_020	Intel CSE Firmware Update	C	☑ ☑ ☑	☑ ☑	I

6.3 Intel CSE Firmware Update

The section serves as a checklist for the environment setup and testing of Intel CSE firmware update and partial (partition) firmware update feature support.

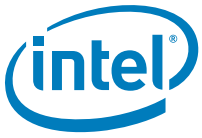
6.3.1 Tools for Testing

A formatted USB Key, the Intel® FWUpdLcl and Intel® MEInfo tools from the Intel CSE firmware kit.

6.3.2 Intel CSE Firmware Update

ID	CS_020
Title	Intel CSE Firmware Update
Requirement	Mandatory - Exempt when upgrade/downgrade support is not yet available in firmware
System	<table><tr><th colspan="2">Form Factor</th><th>System Power Model</th></tr><tr><td>☑ Desktop ☑ Mobile</td><td>☑ Workstation</td><td>☑ Standard ☑ Modern Standby or InstantGo*</td></tr></table>
Method	Manual
Description	Firmware Update settings, as set by the Intel® FIT tool, allow update to the firmware.
Objective	Verify that the Intel® ME firmware can be updated.
Setup	The initial state of the SUT should be S0/MeOn with Host OS running.
Procedure	1. Enter a formatted USB Key into the management console. 2. Browse to an update firmware image on the management console. This may be the latest firmware released by Intel, or an earlier version of the firmware than the firmware currently loaded on the SUT. 3. Place the selected update firmware image on the USB Key. 4. Move the USB Key to the SUT. 5. Run the Intel® FWUpdLcl tool on the SUT with the -save option, to save the current firmware image to the USB Key. 6. Extract the current version of the Intel® ME firmware, using the Intel® MEInfo tool. 7. Run the Intel® FWUpdLcl tool on the SUT to update the firmware to the image on the USB Key. 8. Restart the SUT. 9. Verify the SUT has booted to the Host OS. 10. Extract the new version of the Intel CSE firmware using Intel® MEInfo and ensure that it has changed from the original firmware version. 11. Verify that the new firmware version is correct. 12. Run the Intel® FWUpdLcl tool on the SUT to restore the firmware to the original image extracted earlier from the SUT. 13. Restart the SUT. 14. Verify the SUT has booted to the Host OS. 15. Extract the new version of the Intel CSE firmware using Intel® MEInfo, and ensure that it has been restored to the original firmware version.
Pass Criteria	The test passes, if the firmware update is successful, and the original firmware can be restored for each of the following conditions: • Update to newer version of firmware than what is installed on the SUT. • Downgrade to an older version of firmware than what is installed on the SUT. • It is not allowed to downgrade to an older version of firmware with lower VCN. Depending on the Intel CSE development milestone at which this test is being executed, it may not be possible to fully execute this test with available firmware due to upgrade / downgrade firmware compatibility limitations. In this case, the results for this test become 'Not Available' or 'N/A' untill, such time at which suitable firmware images become available to allow full execution of this test.
References	For details on Intel® ME firmware tools, refer to the *Intel® ME System Tools User Guide*.

§ §



< This page is intentionally left blank >

7 Intel® Platform Trust Technology (Intel® PTT) Compliancy

Intel® Platform Trust Technology (Intel® PTT) is the Intel implementation of TCG TPM 2.0 standard in firmware. For more information about Intel® PTT integration with BIOS, refer BIOS Writers Guide and Intel® PTT Overview documentation.

The purpose of this section is to describe the tests required to verify PTT is functional, main PTT end to end use cases are working and platform meets Windows* 10 requirements for TPM 2.0 support.

The scope of this section is end to end testing and is not intended to provide TPM command level testing.

Note: Intel® Boot Guard testing with Intel® PTT is out of scope of this chapter and should be done as part of Intel® Boot Guard testing.

7.1 Test Environment Setup

- Elkhart Lake Platform with Intel® PTT enabled
- Windows* 10 Professional or Enterprise installed in UEFI mode
- Intel® CSE firmware and Intel® PTT enabled

7.2 Tools for Testing

- **Intel® Platform Enablement Test Suite** - Latest version of the tool is available in the Intel® CSE compliancy kit release. Refer to the Intel® Platform Enablement Test Suite User Guide available in the Intel Compliancy kit for exact instructions on how to load and setup the Intel® Platform Enablement Test Suite software.
- Windows* 10 HLK Testing Environment
- manage-bde.exe (Windows* command line tool for BitLocker Driver Configuration)
- bdehdcfg.exe (Windows* command line tool for BitLocker Drive Encryption)
- makecert.exe (command line tool, part of Windows* 8, Windows* 10 SDK)
- pvk2pfx.exe (command line tool, part of Windows* 8, Windows* 10 SDK)
- CertUtil.exe (Windows* 8, Windows* 10 Command line tool)

7.3 Intel® Platform Trust Technology (Intel® PTT) Compliancy Test Coverage Summary

Platform, Operating System Support, How? Column describes the test methodology.

OS Support: W = Microsoft* Windows*

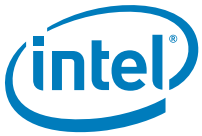
How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M = Manual.

Physical eSPI configuration: S= SPI, U= UFS

Test ID	Test Case Title	PETS Package Name	OS Supported	How?	eSPI Config
PTT_001	CRB Interface Communication Test	Compliance_PTT.xml	W	A	S U
PTT_002	Intel® PTT Windows* 10 Basic Functionality	Compliance_PTT.xml	W	A	S U
PTT_003	TPM Clear and Physical Presence	Compliance_PTT.xml	W	A	S U
PTT_004	Windows* 10 BitLocker Integration	Compliance_PTT.xml	W	A	S U
PTT_005	Windows* 10 BitLocker TPM Protection	Compliance_PTT.xml	W	A	S U
PTT_006	Windows* 10 Virtual Smart Card (VSC) Tests	Compliance_PTT.xml	W	A	S U
PTT_008	Intel® PTT Enable/Disable from BIOS	Compliance_PTT.xml	W	M	S U
PTT_009	Power Transition Testing with Intel® PTT Enabled	Compliance_PTT.xml	W	A	S U
PTT_010	Dictionary Attack Lockout After Coin Battery Removal with EOM Commit	Compliance_PTT.xml	W	M	S U

7.4 CRB Interface Communication Test

Test ID:	**PTT_001**
Test Case Title:	CRB Interface Communication Test
Platform	Elkhart Lake
Mandatory/Optional:	Mandatory ***Note:*** This test uses CRB access and therefore needs to run with disabled driver to ensure elimination of false failures.
Objective:	Verify BIOS is able to successfully send commands to Intel® PTT
Test Pass Criteria:	If TPM_CRB_CTRL_START register returns 0x00 after the duration listed in Table 15 of the TCG specification for the test command sent and before the listed timeout, the TPM command is received by PTT through HCI, the test passes, else fails. Test fails also if a timeout occurs at any other stage. ***Note:*** HCI reference code provides serial output status of whether or not TPM command is received by PTT. Check PttHciReceive function for more details.
Description:	The test confirms that BIOS correctly implements the CRB protocol for communication with Intel® PTT
Procedure:	1. Confirm Intel® PTT is enabled in the image. 2. Disable the Microsoft* TPM driver: From an Elevated Command Window issue the following command: reg add HKLM\SYSTEM\CurrentControlSet\Services\TPM /f /v ImagePath /t REG_EXPAND_SZ /d \SystemRoot\system32\drivers\tpm.sy_ Reboot the system 3. Relinquish locality 0: Write 1 to TPM_LOC_CTRL_0.Relinquish (0xfed40008, bit 1). 4. Request locality 0: Write 1 to TPM_LOC_CTRL_0.RequestAccess (0xfed40008, bit 0). 5. Verify TPM_LOC_STATE_x.locAssigned field (0xfed40000, bit 1) is set to 1 and that TPM_LOC_STATE_x.activeLocality field (0xfed40000, bits 2-4) is set to 000. 6. Write 1 to TPM_CRB_CTRL_REQ_0.cmdReady (0xfed40040, bit 0) 7. Poll TPM_CRB_CTRL_REQ_0.cmdReady every 5 ms for 500 ms until it is 0 8. Verify TPM_CRB_CTRL_STS_0.tpmIdle (0xfed40044, bit 1) is 0 9. Write a TPM command such as TPM2_SelfTest to TPM_CRB_DATA_BUFFER register (0xfed4_0080) 10. Write "1" to the TPM_CRB_CTRL_START register (0xFED4_004C). 11. Poll the TPM_CRB_CTRL_START register (0Xfed4_004C) until its value becomes "0". 12. Write 1 to TPM_CRB_CTRL_REQ_0.goIdle (0xfed40040, bit 1). 13. Poll TPM_CRB_CTRL_REQ_0.goIdle for 500ms until it is 0. 14. Relinquish locality 0: Write 1 to TPM_LOC_CTRL_0.Relinquish (0xfed40008, bit 1). 15. Verify TPM_LOC_STATE_x.locAssigned field (0xfed40000, bit 1) is set to 0 and TPM_LOC_STATE_x.activeLocality field (0xfed40000, bits 2-4) is set to 000. 16. Request locality 0: Write 1 to TPM_LOC_CTRL_0.RequestAccess (0xfed40008, bit 0). 17. Re-enable the Microsoft* TPM driver: From an Elevated Command Window issue the following command: reg add HKLM\SYSTEM\CurrentControlSet\Services\TPM /f /v ImagePath /t REG_EXPAND_SZ /d \SystemRoot\system32\drivers\tpm.sys Reboot the system ***Note:*** For detailed information on how to send a TPM command, refer to the PC client specific platform TPM profile for TPM 2.0



7.5 Intel® Platform Trust Technology (Intel® PTT) Basic Functionality under Windows* 10

Test ID:	**PTT_002**
Test Case Title:	Intel® PTT Basic Functionality Under Windows* 10
Platform	Elkhart Lake
Mandatory/Optional:	Mandatory
Objective:	Windows* 10 can successfully communicate with Intel® PTT
Test Pass Criteria:	No "yellow bang" in device manager, Intel® PTT is the TPM device and all TPM queries return "true"
Description:	Verify Intel® PTT has been enabled on the platform and Intel® PTT is functional on Windows* 10
Procedure:	1. Boot to Windows* 10 UEFI installation 2. Open Device Manager (devmgmt.msc) and verify a "Trusted Platform Module 2.0" device exists in "Security Devices" 3. Open Trusted Platform Module (TPM) Management Page (tpm.msc) 4. Verify Manufacturer Name = **INTC**, TPM Specification Version = **2.0** 5. Verify Status is "The TPM is ready for use." 6. Open an elevated command prompt with admin privileges and enter powershell (type powershell at prompt) 7. Prepare the WMI object for querying Intel® PTT information by typing: *$ptt = get-wmiobject -namespace "root/cimv2/security/microsofttpm" win32_tpm* 8. Check different Intel® PTT parameters by typing the following at the PS prompt: a. $ptt.IsEnabled() b. $ptt.IsActivated() c. $ptt.IsAutoProvisioningEnabled() d. $ptt.IsOwned() e. $ptt.IsReadyInformation()

7.6 Trusted Platform Module (TPM) Clear and Physical Presence

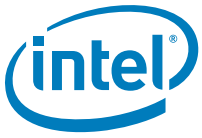
Test ID:	**PTT_003**
Test Case Title:	TPM Clear and Physical Presence
Platform:	Elkhart Lake
Mandatory/Optional:	Mandatory
Objective:	Verify TPM clear and take ownership flows work correctly under Windows* 10 OS and physical presence asserted
Test Pass Criteria:	OS takes ownership of TPM, new/old keys differ
Description:	TPM Clear command erases user data on the TPM. TPM Clear requires BIOS to check for physical presence to authorize the TPM Clear operation. We will save the SrkPublicKey and verify that new/old SRK keys differ after TPM Clear.
Procedure:	1. Save the current SrkPublicKey by performing the following actions: a. Open elevated command prompt and enter PowerShell by typing “powershell” at the prompt and type: b. $ptt = get-wmiobject -namespace "root/cimv2/security/microsofttpm" win32_tpm c. $ret = $ptt.GetSrkPublicKeyModulus() d. $ret.SrkPublicKeyModulus > SrkPubModOld.txt 2. Run “tpm.msc” to open TPM Management Console 3. Click 'Clear TPM…' in the Actions pane on right. 4. In the pop-up window click 'Restart' to invoke TPM Clear flow. 5. Upon reboot, a physical presence authorization message may be displayed (BIOS setting dependent) requiring the user to press a key to authorize the TPM clear or abort. In CRB, F12 will authorize, ESC rejects the operation. 6. Upon booting to Windows*, pop-up window will show up indicating OS is taking ownership of the TPM 7. After ownership operation completes, press OK. 8. Save the new SrkPublicKey by performing the following actions: a. Open elevated command prompt and enter PowerShell by typing “powershell” at the prompt and type: b. $ptt = get-wmiobject -namespace "root/cimv2/security/microsofttpm" win32_tpm c. $ret = $ptt.GetSrkPublicKeyModulus() d. $ret.SrkPublicKeyModulus > SrkPubModNew.txt 9. Compare the old and new keys

7.7 Windows* 10 BitLocker Integration

Test ID:	**PTT_004**
Test Case Title:	Windows* 10 BitLocker Integration
Platform:	Elkhart Lake
Mandatory/Optional:	Mandatory
Objective:	Test BitLocker integration with Intel® PTT
Test Pass Criteria:	All system boots complete successfully and OS loads
Description:	BitLocker uses Intel® PTT to store and retrieve keys securely, in addition Windows* BitLocker confirms system components did not change by checking system load measurements saved to TPM. The test will verify BitLocker can be activated, BitLocker can encrypt, decrypt, and restart encryption after reboot.
Procedure:	1. In elevated permissions command line run: "bdehdcfg.exe –driveinfo" and check system drive is configured to support BitLocker 2. Set BitLocker to use TPM for measuring boot devices in Windows* Group Policy by: a. Run "gpedit.msc" to open Group Policy Editor b. Open "Local Computer Policy" > "Computer Configuration" > "Administrative Templates" > "Windows Components" > BitLocker Drive Encryption" > "Operating System Drives" c. On the right pane double click "Configure TPM platform profile for native UEFI firmware configuration" d. Check the enabled radio button. Verify PCR 0, PCR2, PCR4 and PCR11 are checked in the "Options" pane. e. Click apply and OK. f. Commit the group policy change by typing "gpupdate /force" in an elevated command prompt ***Note:*** This action is required once per OS installation 3. Set up tpm as a bitlocker protector with recovery password and turn-on BitLocker by typing the following at the command prompt a. manage-bde -protectors -add c: -tpm b. manage-bde -protectors -add c: -rp 000000-000000-000000-000000-000000-000000-000000-000000 c. manage-bde -on c: d. shutdown –r –t 0 4. After OS completes reboot, verify no error messages displayed. Wait for "Encryption in Progress" notification or type "manage-bde -status" to check on encryption status 5. After encryption reaches 3%, restart system, and verify encryption continues without error message after reboot completes. 6. Turn off BitLocker by typing "manage-bde –off c:" at the command line, decryption process should start 7. After decryption process ends, reboot and verify system boots into OS without error message. BitLocker should be off

7.8 Windows* 10 BitLocker TPM Protection

Test ID:	**PTT_005**
Test Case Title:	Windows* 10 BitLocker TPM Protection
Platform:	Elkhart Lake
Mandatory/Optional:	Mandatory
Objective:	Verify BitLocker is using Intel® PTT for TPM protection
Test Pass Criteria:	BitLocker completes drive encryption successfully and reboots. System displays BitLocker recovery screen after choosing Disable Intel® PTT or Clear TPM in BIOS setup.
Description:	When BitLocker is set to use TPM protection, BitLocker will enter recovery mode if any protected component changed during boot. By disabling Intel® PTT, we will check BitLocker is indeed using TPM protection.
Procedure:	1. Encrypt the OS drive using BitLocker with TPM protection (follow instructions in PTT_004 steps 1 through 5, and wait till drive encryption reaches 3%) 2. Run manage-bde –status and verify drive is "protected" 3. Create a measured boot failure in order to trigger Bitlocker Recovery a. In BIOS, choose disable Intel® PTT or send a TPM_Clear command. ***Note:*** Clearing TPM by means of the OS will disable Bitlocker and will not prompt the user for his recovery password. The TPM must be cleared by the BIOS. b. System should boot into BitLocker recovery screen. Provide the recovery password to continue boot. c. Verify boot completes successfully 4. Disable BitLocker by typing "manage-bde –off c:" at the command line, decryption process should start 5. After decryption process ends, reboot and verify system boots into OS without error message. BitLocker should be off



7.9 Windows* 10 Virtual Smart Card Tests

Test ID:	**PTT_006**
Test Case Title:	Windows* 10 Virtual Smart Card (VSC) Tests
Platform:	Elkhart Lake
Mandatory/Optional:	Mandatory
Objective:	Intel® PTT can be used to support VSC use case
Test Pass Criteria:	VSC created successfully, certificate can be loaded and is persistent across reboot. VSC can be removed after key is deleted
Description:	Virtual Smart Card is a new Microsoft* use case for TPMs. More information on VSC can be found on Microsoft* web site. This test verifies a VSC can be created and certificate installed so VSC is accessible
Procedure:	1. Create a VSC running the following command on an elevated command line: tpmvscmgr.exe create /name TPM2VSC /adminkey random /PUK default /pin default /generate 2. Verify that TPM2VSC smart card reader was created in "Smart card readers" in device manager 3. Restart Windows*, and check the device is not yellow banged in device manager 4. Create and import a self-signed certificate into the VSC a. Ensure the following registry keys exist under [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Microsoft Base Smart Card Crypto Provider]: • "AllowPrivateSignatureKeyImport"=DWord:00000001 • "AllowPrivateExchangeKeyImport"=DWord:00000001 b. Open an elevated command prompt c. Type: MakeCert.exe -sky exchange -r -n "CN=TPM2VSCCert" -pe -a sha1 -len 2048 -ss My -m 36 -sv "TPM2VSCCert.pvk" "TPM2VSCCert.cer" d. When requested, create a password. When asked for the password, provide the password created (for this example, using "123" as the password) e. Convert certificate to PFX format using the following command: pvk2pfx.exe -pvk "TPM2VSCCert.pvk" -pi 123 -spc "TPM2VSCCert.cer" -pfx "TPM2VSCCert.pfx" –f f. Import the certificate into the smart card using the following command: CertUtil.exe -p 123 -csp "Microsoft Base Smart Card Crypto Provider" -pin 12345678 -importpfx TPM2VSCCert.pfx AT_KEYEXCHANGE 5. Verify import was successful by examining the certificate in the VSC using the following command: CertUtil.exe -scinfo –pin "12345678". Window allowing to view the certificate will pop up, click OK to close 6. Restart the platform, and run step 5 again, to verify certificate persists after reboot 7. Remove the key from the VSC using the following commands a. Retrieve the name of the container to use by typing: CertUtil.exe -key -csp "Microsoft Base Smart Card Crypto Provider" -pin "12345678" -v -privatekey –user b. Use the container name returned in the previous command prefixed to the "[Default Container]" and replace the text in bold: CertUtil.exe -delkey -csp "Microsoft Base Smart Card Crypto Provider" -pin 12345678" -v -privatekey **"TPM2VSCCert-0d6e6c94-9bd6-4640-aa-63900"** 8. Destroy the VSC by running: `TpmVscMgr.exe destroy /instance ROOT\SMARTCARDREADER\0000`, making sure to use the correct index of the smartcard created

7.10 Intel® Platform Trust Technology (Intel® PTT) Disable/Enable from BIOS

Test ID:	**PTT_008**
Test Case Title:	Intel® PTT Disable/Enable from BIOS
Platform:	Elkhart Lake
Mandatory/Optional:	Mandatory
Objective:	Ensure BIOS can enable and disable Intel® PTT successfully and that BIOS clears the TPM during disable
Test Pass Criteria:	When Intel® PTT is disabled; Intel® PTT does not show up in TPM management console. (It's possible for dTPM to show up pending on platform design).
Description:	BIOS may implement option to disable/enable Intel® PTT, or switch between Intel® PTT and a discrete TPM 1.2
Procedure:	1. Boot to OS, verify PTT_002 passing. 2. Reboot, enter BIOS and disable Intel® PTT through BIOS 3. Boot to Windows*, enter TPM Management Console (tpm.msc) and verify that either TPM is not available, or if TPM is available it is not Intel® PTT 4. Reboot, enter BIOS and enable Intel® PTT through BIOS 5. Boot to OS, verify PTT_002 passing ***Note:*** Intel® PTT enable/disable interface in BIOS is dependent on implementation and therefore not described

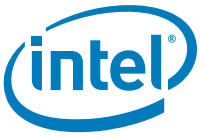
7.11 Intel® Platform Trust Technology (Intel® PTT) and Power Flows

Test ID:	**PTT_009**
Test Case Title:	Power Flow Testing
Platform:	Elkhart Lake
Mandatory/Optional:	Mandatory
Objective:	Verify Intel® PTT does not interfere with system power operations
Test Pass Criteria:	All power flow tests pass, BitLocker does not enter into recovery mode
Description:	System with Intel® PTT enabled must pass all platform power flow testing. Intel® PTT must also be able to support all power flows when BitLocker is enabled and using Intel® PTT as a protector
Procedure:	1. Perform all platform power flow tests with Intel® PTT enabled 2. Encrypt the OS drive using BitLocker with TPM protection (follow instructions in PTT_004 steps 1 through 5, and wait till drive encryption reaches 3%) 3. Perform the following power transitions during encryption phase and after encryption has reached 3%: a. OS Restart b. OS ShutdownPower up c. Hibernation Resume d. Cold Reset (boot to internal EDK shell and type mm cf9 e -io) e. G3 (complete power off) f. Connected Standby (Windows* 8.1 CS) 4. After each Flow – test/verify PTT functionality (in S0) – run PPT_002

7.12 Dictionary Attack Lockout after Coin Battery Removal with EOM Commit

Test ID:	**PTT_010**
Test Case Title:	Dictionary Attack Lockout Mechanism with coin battery removal
Platform:	Elkhart Lake
Mandatory/Optional:	Optional
Objective:	Allows OEM to validate the dictionary attack scenario after first coin battery removal, causing the counters to be reset. ***Note:*** This test is only for designs with RTC powered by coin cell battery.
Test Pass Criteria:	Intel® PTT will not allow access to user data (VSC) during lockout period post coin battery removal ***Note:*** In this test Field Programmable Fuses (FPF) will be blown on every battery removal and there is no recovery for it. Only select few processors to be used for this test and track them.
Description:	Intel® PTT keeps monotonic counters for Dictionary Attack (DA) under RTC power well. When RTC power is lost, Intel® PTT will enter lockout period to avoid Dictionary Attack for 2 hours. This is only after the coin battery has been removed 10 times and after EOM. Before that, Intel® PTT will not enter the lockout period of 2 hours. ***Note:*** During the 2 hour lockout period, no other Intel® PTT tests can be executed; even if correct credentials are provided. Execution of this test does not impact other non-Intel® PTT related testing. ***Note:*** This test can be run only once on a specific part. After this test is run, all FPF bits related to the feature will be blown. With such parts, test will consistently enter dictionary attack scenario after every RTC clear operation. **WARNING:** This flow is irreversible, the part will be permanently fused causing every RTC clear to cause a 2 hour TPM lockout.
Procedure:	1. System must post EOM procedure, as DA lockout will not occur during manufacturing mode 2. Set up a VSC with certificate (Instructions can be found in test PTT_006 steps 1 through 6) 3. Shutdown system, and perform RTC clear operation by removing all power and RTC battery from the board and close the RTC jumper. Repeat this procedure 11 times. 4. Return RTC battery and power, boot system to Windows* 5. Try to view the certificate in VSC by running: CertUtil.exe -scinfo -pin "12345678". 6. The command should fail due to Dictionary Attack lockout 7. Check the configured lockout configuration [3 min/10 tries OR 2 hours/ 32 tries] 8. According to the configuration (3 min/2 hours), wait for lockout to pass, and try again, it should be possible to access the certificate 9. Remove the certificate and VSC (Instruction can be found in test PTT_006 steps 8 and 9) ***Note:*** At step#3, the Intel® PTT is expected to enter a lockout period to avoid Dictionary Attack for 2 hours. This period cannot be adjusted.

§ §



< This page is intentionally left blank >

8 Intel® Boot Guard Compliancy

Boot Guard is an Intel platform boot integrity protection technology. Boot Guard can help protect the platform boot integrity by preventing execution of unauthorized boot block. With Boot Guard, the OEM can create a platform boot policies such that invocation of an unauthorized (or compromised) boot block will trigger the platform protection per the OEM policies. Based in the hardware, Boot Guard will also extend the trusted boundary of the platform boot process down to the hardware. A benefit of this protection is that Boot Guard can help OEM maintains platform integrity by preventing reuse of the OEM hardware to run unauthorized software stack.

8.1 Scope

This section describes a validation strategy for Boot Guard Compliancy. This chapter is intended for validation purposes. The objective is to provide validation professionals with additional insight into Boot Guard by highlighting validation considerations. This chapter is not a technology overview and does not replace the existing Boot Guard collateral. The reader is expected to be familiar with Boot Guard and to use this document as a validation supplement to develop his own validation plan.

8.2 Prerequisite

8.2.1 Tools Supported

This Boot Guard evaluation plan documented in this chapter requires the following components and tools for execution.

Tool/Component	Revision	Comments
Manifest Generation Tool (MEU.exe)	CSE Firmware Kit with Boot Guard Support	MEU tool is required to sign BIOS components and generate manifests.
FIT	CSE Firmware Kit with Boot Guard Support	FIT is required to define the Boot Guard Boot Policies (persistent policies). Available on VIP
MEInfo	CSE Firmware Kit with Boot Guard Support	MEInfo is required to confirm Boot Guard Policies.
FPT (Flash Programming Tool)	CSE Firmware Kit with Boot Guard Support	FPT will be used to commit the boot guard related values to IFP, read and display the values from IFP
MEManuf	CSE Firmware Kit with Boot Guard Support	To compare the flash/IFP Boot Guard values and display the result
TxtBtgInfo	0.7.10 or higher	The tool will verify the integrity of the OEM Key manifest and the Boot Policy Meta data File Extension

8.2.2 Boot Media Support

The below tests are applicable for all boot Media Devices (SPI):

8.3 Boot Guard Test Coverage Summary

Note: Profile 1 and profile 2 support has been deprecated. Only Profile 0: NO_FVME, Profile 3: VM, Profile 4: FVE and Profile 5: FVME is supported.

(Profile 3 is only supported for pre-production/debug configuration and its not supported after end of manufacturing has been committed.)

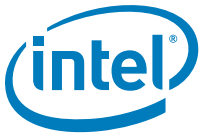
(Verified only refers to profile 4 as unlike profile 5 - measured and verified it only has verified enabled)

(Profile 0: NO_FVME should be set when disabling Boot Guard)

How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M= Manual.

Network Factor: LAN = systems with LAN interface and test is performed using LAN interface, WLAN = systems with WLAN interface and test is performed using the WLAN interface.

Test ID	Test Case Title	Target OS	How?
ME_BtG_001	Platform Boot with Boot Guard Disabled (using FPF Emulation)	W	M
ME_BtG_002	Successful verified only Boot to OS (using FPF Emulation)	W	M
ME_BtG_003	Unsuccessful verified only boot (using FPF Emulation)	W	M
ME_BtG_004	Successful VM (Verified Measured) Boot to OS (using FPF Emulation)	W	M
ME_BtG_005	Unsuccessful VM (Verified Measured) Boot to OS (using FPF Emulation)	W	M
ME_BtG_006	Successful VM (Verified Measured) Boot to OS using HW FPFs	W	M



8.4 ME Boot Guard 001

Test ID:	**ME_BtG_001**
Test Case Title:	Platform Boot with Boot Guard Disabled (using FPF Emulation)
Objective:	Objective of the test is to verify that the platform boots with Boot Guard Disabled.
Test Pass Criteria:	Platform successfully boots to OS and verified or measured boot should not have executed. MEinfo output for Boot guard profiles under the "ME" column should show Verified Boot and Measured Boot Disabled. All other settings should be as per the configuration done via FIT tool.
Description:	In this test case, boot guard flow for verification and measurement of IBB will not be executed during the platform boot process.
Windows* Procedure:	**Prerequisite** • Use FIT tool to create the full image with relevant Boot Guard legacy provisioning in the Platform Protection tab as per the details in the Firmware Bring up guide • Run the TxtBtgInfo tool with the below command line option to verify the integrity of the OEM Key manifest and the Boot Policy Metadata file extension — TxtBtgInfo_v0.x.x.efi -f <IFWI Image> — The test results should show pass for KM and BPM (would not be present in case of a legacy profile) **Prepare the SUT** 1. Provision the SUT (NVAR if this is development system) to Legacy profile. Refer the CSE Firmware Bring Up Guide for information on the Boot Guard related FIT options and settings 2. Boot the platform to OS or EFI shell 3. Run the MEinfo tool to check the profile settings.

8.5 ME Boot Guard 002

Test ID:	**ME_BtG_002**
Test Case Title:	Successful verified only Boot to OS (using FPF Emulation)
Objective:	This test verifies that the SUT has all the required components: hardware, firmware and BIOS. Additionally, the SUT has been correctly provisioned in manufacturing for the platform to boot with the Boot Guard for IBB verification
Test Pass Criteria:	1. Platform boot should be successful without any hangs or errors. 2. MEInfo output should be as per the boot guard profile configured. It should show Verified only enabled and check for the fields under "FPF" column for FPF contents and "ME" for NVAR contents. Ensure that these matches with what was provisioned during the image creation process.
Description:	In this Test case, Boot guard will authenticate and load the IBBL and perform a successful verification of the SUT (IBB) or Initial Boot Block, IBB would in turn authenticate the rest of the BIOS components.
Windows* Procedure:	**Prerequisite** • Use the MEU tool for signing the BIOS components and generating the manifest (BPM and KM) • Stitch the BIOS components (IBBL, IBB, OBB) together using the MEU.exe tool to generate one single BIOS ROM image • Run the BtGinfo tool with the below command line option to verify the integrity of the OEM Key manifest and the Boot Policy Metadata file extension — TxtBtgInfo_v0.x.x.efi -n <IFWI Image> — The test results should show pass for both KM and BPM • Use FIT tool to create the full image with relevant Boot Guard provisioning in the Platform Protection tab as per the Firmware Bring up guide. **Prepare the SUT** 1. Provision the SUT (NVAR if this is development system) to Verified only profile. Per testing objective Refer the CSE Firmware Bring Up Guide for information on the Boot Guard related FIT options and settings 2. Flash the image and boot the platform to OS 3. Run the MEinfo tool to check the profile settings 4. Check the platform behavior as per the details in the Test Pass Criteria.

8.6 ME Boot Guard 003

Test ID:	**ME_BtG_003**
Test Case Title:	Unsuccessful verified only Boot to OS (using FPF Emulation)
Objective:	To verify that the platform will fail to boot if verified boot is enabled if there is an IBB corruption or ACM
Test Pass Criteria:	Platform should shutdown or enter the DnX mode
Description:	In this Test case, Boot guard will fail the verification of the SUT (IBB) or Initial Boot Block and prevent the platform from booting
Windows* Procedure:	**Prepare the SUT** 1. Provision the SUT (NVAR if this is development system) to Verified only profile. Per testing objective Refer the CSE Firmware Bring Up Guide for information on the Boot Guard related FIT options and settings 2. Provision an incorrect BootPolicyManifest Hash Public Key while creating the image or change signature of the manifests using the MEU tool. ***Note:*** To provision an incorrect "Boot Policy Manifest" key for this test: a. Change the build settings in the FIT tool for the field "Verify manifest signing keys against the OEM key Manifest" to "No" to avoid build failure with an incorrect value for the negative test. b. Using MEU, create OEM Key Manifest binary with corrupted key hash binary for BootPolicyManifest (refer to Signing and Manifesting guide for how to create OEM KM binary). i. May corrupt the key hash binary using hex editor. 3. Install the targeted OS if not already installed on the SUT and try booting the platform.

8.7 ME Boot Guard 004

Test ID:	**ME_BtG_004**
Test Case Title:	Successful VM(Verified/Measured) Boot to OS (using FPF Emulation)
Objective:	This test verifies that the SUT has all the required components: hardware, firmware and BIOS. Additionally, the SUT has been correctly provisioned in manufacturing for the platform to boot with the Boot Guard for IBB verification and Executes Measured only boot.
Test Pass Criteria:	1. Platform successfully boots to OS 2. Check for the fields under "ME" for NVAR contents, verified boot and Measured Boot should be enabled. Ensure that these matches with what was provisioned during the image creation process.
Description:	In this test Platform would boot with IBB successful IBB verification and measurement of the BIOS components

Test ID:	ME_BtG_004
Windows* Procedure:	**Prerequisite** • Use the MEU tool for signing the BIOS components and generating the manifest (BPM and KM) • Stitch the BIOS components (IBBL, IBB, OBB) together using the MEU.exe tool to generate one single BIOS ROM image • Run the BtGinfo tool with the below command line option to verify the integrity of the OEM Key manifest and the Boot Policy Metadata file extension — TxtBtgInfo_v0.x.x.efi -n <IFWI Image> — The test results should show pass for both KM and BPM • Use FIT tool to create the full image with relevant Boot Guard provisioning in the Platform Protection tab as per the Firmware Bring up guide. **Prepare the SUT** 1. Provision the SUT (NVAR if this is development system) to VM profile. Per testing objective Refer the CSE Firmware Bring Up Guide for information on the Boot Guard related FIT options and settings 2. Boot the system to OS 3. Run the MEinfo tool to check the profile settings.

8.8 ME Boot Guard 005

Test ID:	ME_BtG_005
Test Case Title:	Unsuccessful VM(Verified/Measured) Boot to OS (using FPF Emulation)
Objective:	To verify that the platform will fail to boot if verified boot and measured boot is enabled under the condition when there is an IBB corruption or ACM
Test Pass Criteria:	Platform should shutdown or enter the DnX mode
Description:	In this test case boot guard will perform an unsuccessful verification and measurement of the SUT (System Under test) Initial Boot Block (IBB). Upon verification failure platform boot will be prevented and System will enter the recovery mode.
Windows* Procedure:	**Prerequisite** • Use the MEU tool for signing the BIOS components and generating the manifest (BPM and KM) • Stitch the BIOS components (IBBL, IBB, OBB) together using the MEU.exe tool to generate one single BIOS ROM image • Run the BtGinfo tool with the below command line option to verify the integrity of the OEM Key manifest and the Boot Policy Metadata file extension — TxtBtgInfo_v0.x.x.efi -n <IFWI Image> — The test results should show pass for both KM and BPM • Use FIT tool to create the full image with relevant Boot Guard provisioning in the Platform Protection tab as per the Firmware Bring up guide. **Prepare the SUT** 1. Provision the SUT (NVAR if this is development system) to VM profile. Per testing objective Refer the CSE Firmware Bring Up Guide for information on the Boot Guard related FIT options and settings 2. Flash the Intel® CSE firmware and BIOS image that are Boot Guard enabled with VM profile 3. Corrupt the Manifests or Hash keys to create a fail scenario. (E.g. enter the wrong public hash key while image creation) 4. Try booting the platform to OS.

8.9 ME Boot Guard 006

Test ID:	ME_BtG_006
Test Case Title:	Successful VM(Verified/Measured) Boot using HW FPFs
Objective:	This test verifies that the SUT has all the required components: hardware, firmware and BIOS. Additionally, the SUT has been correctly provisioned in manufacturing for the platform to boot with the Boot Guard for IBB verification and executes Verified and Measured boot using the values from the FPFs instead of FPF mirror
Test Pass Criteria:	1. Platform successfully boots to OS 2. Run the MEinfo tool to check the profile settings. It should show Verified and Measured Boot enabled under "FPF" column. And all the other settings for Boot Guard should be reflected as per the configurations done using FIT tool.
Description:	In this test case Boot Guard will perform the feature testing using the values from the FPFs (i.e. accessing the profile values from the FPFs instead of the flash variables, NVARs)
Windows* Procedure:	**Prerequisite** • Use the MEU tool for signing the BIOS components and generating the manifest (BPM and KM) • Stitch the BIOS components (IBBL, IBB, OBB) together using the MEU.exe tool to generate one single BIOS ROM image • Run the BtGinfo tool with the below command line option to verify the integrity of the OEM Key manifest and the Boot Policy Metadata file extension — TxtBtgInfo_v0.x.x.efi -n <IFWI Image> — The test results should show pass for both KM and BPM • Use FIT tool to create the full image with relevant Boot Guard provisioning in the Platform Protection tab as per the Firmware Bring up guide • Perform this test ONLY when all the above tests (ME_BtG_001 to ME_BtG_005) has passed using the profile values from the FPF mirror or NVARs. ***Note:*** The profiles selected to be committed into FPFs will become the final profile which cannot be altered later. Take care of the prerequisites before proceeding further with the test **Prepare the SUT** 1. Install the targeted OS if not already installed on the SUT 2. Provision the SUT with a desired Boot Guard Profile (Say VM) per testing objective. Refer the CSE Firmware Bring Up Guide for information on the Boot Guard related FIT options and settings along with the Signing and Manifest Guide 3. Perform the step to commit the Boot guard profile value to FPFs. This will be done automatically after CSE manufacturing mode is disabled (during the global reset from FPT –closemnf or first boot for pre-lock image) if FW and Si are both production, or done by means of a specific FPF MEI command (If combination of FW and Si is Pre-production). Below commands can be used for FPF commit (Also refer to the CSE tools user guide for the commands usage) "FPT –FPFs" - To retrieve the FPF names "FPT –CLOSEMNF" – To commit all of FPF values FPF HW Boot the Platform to the desired OS.

§ §

< This page is intentionally left blank >

9 Signing, Manifesting, and Secure Tokens

This chapter includes tests to verify that OEMs are able to add OEM signed components to the platform image, create Secure Tokens for Debug, and successfully inject them into the platform. It also verifies that OEM unlock token works on the platform.

Secure Tokens are only supported on platforms with a other OEM signed components.

The tests in this chapter are only relevant for OEMs who wish to sign OEM components in the platform image.

9.1 Test Environment Setup

Signing and manifesting documentation can be found in CSE FW kit that details usage of signing the tokens.

9.2 Tools for Testing

- PFT (Platform Flash Tool): Tool used for DnX mode, and token creation/injection. Tool can be found in latest CSE FW kit.
- OpenSSL: Freeware, can be found in Open source community.
- FIT (Flash Image Tool): Tool used to stitch FW image, can be found in CSE FW kit.
- FPT (Flash Programming Tool): Tool used to burn images on SPI platforms, and set EOM state.

9.3 Signing, Manifesting, and Secure Tokens Test Coverage Summary

Platform, Operating System Support, How? Column describes the test methodology.

OS Support: W = Microsoft* Windows*, AOS = Android OS, L = Linux

How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M = Manual.

Test ID	Test Case Title	PETS Package Name	OS Supported	How?
SIGN 01	Image creation with OEM signed component	N/A	W WI AOS **L**	M
SECTOK 02	Debug Token	N/A	W WI AOS **L**	M

9.4 Non-Signed Image Creation

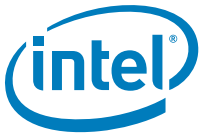
Test ID:	SIGN 01
Test Case Title:	Image Creation with OEM Key manifest
Objective:	This test verifies that OEM Km is signed correctly, and platform boots with empty OEM KM in image.
Test Pass Criteria:	Platform boots and image with signed OEM KM loads correctly. This indicates the signing process is correct.
Description:	OEM will sign OEM KM and add it to the image.
Procedure:	**Manual procedure.** 1. Create pairs of 3K keys for signing OEM KM, using OpenSSL. 2. Use MEU from project kit to generate the OEM KM file. 3. Edit the file to indicate the OEM KM will be empty. (no keys included) 4. Manifest and sign the OEM Key Manifest using MEU from kit (which uses SHA-384). 5. Enter the hash of the OEM Key Manifest key and the OEM Key Manifest binary into FIT, and then use FIT to build an IFWI image including the OEM Key Manifest. 6. Burn the IFWI image to the platform. 7. Verify that the platform boots to the OS. * Details of the procedures above are in the Signing and Manifesting Guide

9.5 Debug Token

Test ID:	SECTOK 02
Test Case Title:	Debug Tokens
Objective:	This test verifies that OEMs are able to create Secure Tokens for Debug, and successfully inject them into the platform. It also verifies that OEMs are able to enter the hash of the token public key into the OEM Key Manifest, and build an image with this manifest, such that the platform will recognize the injected token.
Test Pass Criteria:	Platform is in OEM Unlock State
Description:	OEM will create a token. The public key hash will be entered into the OEM Key Manifest, which will be included in the IFWI image. The token will be injected into the platform using DnX. Platform moves to OEM Unlock Status
Windows* Procedure:	**Manual procedure.** 1. Create a pair of keys for the Debug Token, for example, using OpenSSL. Details of procedure are in the Elkhart Lake Secure Tokens guide found in latest CSE FW kit. In order to use the Intel® Platform Flash tool to create tokens, the Private key and the password used to create this key should be entered in the Intel® Platform Flash tool under Security tab (on the top) -> General Settings as Certificate and password respectively. 2. Enter the public key hash into the OEM Key Manifest's field for OEMUnlockTokens. Details of procedure for creating the hash are in the Elkhart Lake Signing and Manifesting Guide, chapter 3, and details for entering the hash into the OEM Key Manifest are in chapter 5. 3. Use MEU to manifest and sign the OEM Key Manifest. Details of procedure are in the Elkhart Lake Signing and Manifesting Guide, chapter 5. 4. Use FIT to build an IFWI image including the OEM Key Manifest. Details of procedure are in the Elkhart Lake Signing and Manifesting Guide, chapter 5. a. In order to create and sign an OEM Unlock token, use the Intel® Platform Flash Tool ensuring to set the OEMUnlockEnabled knob to OEMUnlockEnabled, and the ISH GDB Debug knob to "enabled" — follow instructions in the Elkhart Lake Secure Token guide. b. To stitch token within the IFWI image, — Use FIT also to add token in image. — Burn the IFWI image to the platform, and use FPT –EOM to close manufacturing state. c. If you choose NOT to stitch within IFWI, then continue to inject token via Intel® Flash Programming tool OR Intel® Platform Flash Tool if you are using DnX APIs (Refer to Elkhart Lake Secure Token Guide for instructions) 5. Verify that platform functionality is (Orange) unlocked, and available for debugging. Ensure the following are done: a. After injecting token via DnX or stitched in image by FIT, boot platform with DCI enabled b. Connect Lauterbach to capture NPK messages (using NPK decoder released in compliance kit) c. Initiate warm reset d. Verify NPK log contains the following message: "Accept secure token".

§ §

< This page is intentionally left blank >



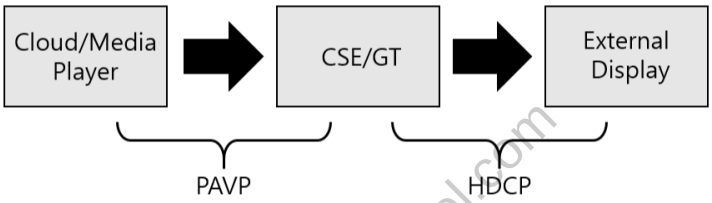
10 Protected Media Playback

10.1 Overview

Protected Media Playback is supported by Intel® CSE Firmware. Intel®CSE employs the following content protection mechanism to safe guard premium content form copy:

a. Intel® Protected Audio Video Path

b. Intel® High-bandwidth Digital Content Protection



The Protected Audio/Video Path (PAVP) is an Intel-specific collection of content protection features in the Intel "Gen" graphics products. The purpose of PAVP is to support premium content video playback including Blu-ray discs and provide a protected path from the media player application to the GPU HW.

Protection of the data as it leaves the GPU and goes to an external display is typically done using industry standard HDCP.

10.2 Scope

This chapter describes a validation strategy for protected content Protected Media Playback. This chapter is intended for validation purposes. The objective is to provide validation professionals with additional insight into Media Playback protection offered by Intel® CSE by highlighting validation considerations. This chapter is not a technology overview. The reader is expected to be familiar with Protected Media Playback or Content Protection and to use this document as a validation supplement to develop his own validation plan.

10.3 Prerequisite

This Protected Media Playback evaluation plan documented in this chapter requires the following components and tools for execution.

Intel® Flash Image Tool (fit.exe)

Intel® Flash Programming Tool (Intel® FPT) - is available in Windows* 32-bit (fptw.exe), Windows 64-bit (fptw64.exe) operating systems, EFI 32-bit and EFI 64-bit.

10.4 Test Environment Setup

The System under Test (SUT) is to be configured in manual configuration mode a with wired LAN or wireless LAN dynamic IP address. The DHCP server connecting the SUT and Management Console (MC) must be configured to ensure that the wired LAN and wireless LAN addresses reside on separate subnets. The MC could be a laptop or desktop system running a version of Windows* supported by PETS. The network configuration consists of a hub or switch, network cables, and a wireless Access Point (AP).

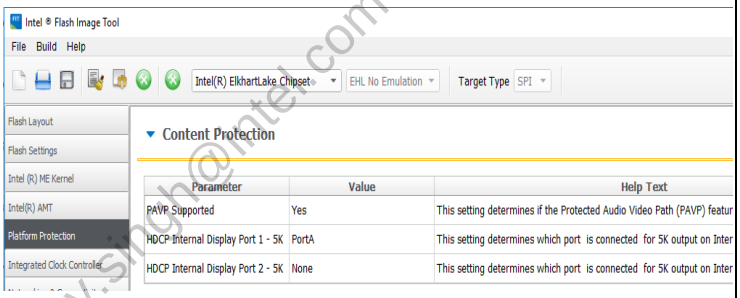
10.5 Media Playback Test Coverage Summary

Platform, Operating System Support, How? Column describes the test methodology.

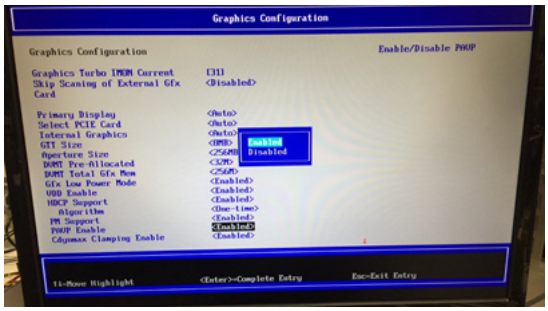
OS Support: W = Microsoft Windows *, WI = Microsoft* Windows* InstantGo, AOS = Android OS, L = Linux

How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M = Manual.

Test ID	Test Case Title	PETS/Manual	OS Supported
Media_001	Verify default configuration settings for Protected Audio Video Path [PAVP] in Firmware Image Tool [FIT]	Manual	W WI AOS L
Media_004	Verify PAVP Enabled in BIOS (***Only if the SUT BIOS menu displays PAVP Mode***)	Manual	W WI AOS L

Test ID:	**Media_001**
Test Case Title:	Verify default configuration settings for Protected Audio Video Path [PAVP] in Firmware Image Tool [FIT]
Mandatory/Optional:	Mandatory
Description:	Intel® CSE initiates PAVP secure session in firmware for key exchange and encryption for Content from Media player or cloud. PAVP can be enabled or disabled using FIT Tool. In this test we verify the PAVP is enabled in the SUT SPI image using FIT.
Objective:	Verify if the PAVP control in Intel® FIT are set correctly
Procedure:	1. Open customer image in FIT tool 2. Got to Platform Protection tab 3. Verify and ensure if the 'PAVP Supported Parameter' is set to 'Yes'  ***Note:*** Above picture is for Window FIT
Test Pass/Fail Criteria:	Test passes is FIT PAVP parameter is set to 'Yes' when we open SPI image in FIT.

Test ID:	**Media_004**
Test Case Title:	Verify PAVP Enabled in BIOS
Mandatory/Optional:	Mandatory (Only if the SUT BIOS menu displays PAVP Mode)
Description:	PAVP can be configured in the BIOS. In this test we will verify what the PAVP mode is enabled in SUT BIOS.
Objective:	Verify PAVP configuration in BIOS

Test ID:	**Media_004**
Procedure:	1. Boot system to BIOS menu 2. Navigate in your BIOS menu where you have PAVP Option [e.g. in Intel BIOs goto - Intel Advance Menu->System Agent (SA) Configuration->Graphics Configuration-> PAVP Enable 3. Verify the PAVP mode setup 
Test Pass/Fail Criteria:	Test passes if we PAVP is enabled in SUT BIOS.

§ §

11 Intel® Dynamic Application Loader (Intel® DAL)

11.1 Introduction

Intel® Dynamic Application Loader (Intel® DAL) is an Intel® CSE infrastructure for applications.

The following table documents compliancy tests to verify Intel® Dynamic Application Loader (Intel® DAL) is working on the platform.

This Test plan is targeted at all OEMs.

11.2 Test Environment for the Intel® Dynamic Application Loader (Intel® DAL)

Note: No OEM implementation is required on the board/BIOS or EC level. Intel® CSE should be set to Enabled in FIT when creating the firmware image.

The Management console could be a laptop or a desktop with a version of Windows*, Linux* and Android* that are supported by Intel® Platform Enablement Test Suite. The network to use is a hub/switch and network cables.

The Intel® DAL tests should not be conducted in Windows* Server 2008 as Intel® DAL currently does not supports this OS.

Note: DAL Applet needs to be signed with RSA 3K due to CSE signing method upgrade.

11.2.1 Tools for Testing

Intel® Platform Enablement Test Suite—Latest version of the tool from the Intel® CSE Compliancy kit release. Refer to the Intel® Platform Enablement Test Suite (Intel® PETS) user guide available in the Intel® Compliancy kit for exact instructions on how to load and setup the Intel® Platform Enablement Test Suite software.

Package DAL.xml should be loaded to Intel® PETS in order to compete the tests in this section

11.2.2 Verify Needed Software is Installed on Host

The following software components need to be available in the platform OS:

Intel® MEI Driver:
This is the interface used for communication between the host OS components and the Intel® CSE components (included in the general Intel® CSME installer kit).

Intel® Dynamic Application Loader (Intel® DAL) host software components:
Exposes an API that allows communication between the host client and the application (included in the general Intel® CSE installer kit)

11.3 Intel® Dynamic Application Loader (Intel® DAL) Test Coverage Summary and Details

Test ID	Test Case Title	PETS/Manual	Form Factor
DAL_001	Intel® DAL applications cleanup	PETS	Mobile
DAL_002	Intel® DAL test application installation and load	PETS	Mobile
DAL_003	Intel® DAL communication channel exercise	PETS	Mobile
DAL_004	Intel® DAL Application signing by OEM	PETS	Mobile

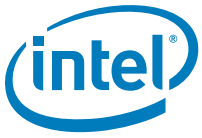
Test ID:	**DAL_001**
Test Case Title:	Intel® DAL applications cleanup
Mandatory/Optional:	Mandatory
Firmware SKU:	Mobile SKU
Description:	Intel® DAL applications cleanup mechanism test
Objective:	To test that the Intel® Dynamic Application Loader (Intel® DAL) clean-up mechanism works properly, and no application is currently running in Intel® DAL
Procedure:	Start test DAL_001 in the Intel® Platform Enablement Test Suite from management console. **Intel® Platform Enablement Test Suite performs the following steps:** 1. Confirm the Intel® Dynamic Application Loader (Intel® DAL) is enabled in firmware. 2. Confirm needed Host software components are available (Intel® MEI driver and Intel® Dynamic Application Loader (Intel® DAL) host software). 3. Perform cleanup of all Intel® DAL applications.
Test Pass/Fail Criteria:	All steps return the value "Passed"

Test ID:	**DAL_002**
Test Case Title:	Intel® DAL test application installation and load
Mandatory/Optional:	Mandatory
Firmware SKU:	Mobile SKU
Description:	Intel® DAL test application is installed and loaded, verifying basic functionality of Intel® DAL applications execution capability.

Test ID:	**DAL_002**
Objective:	To test that the Intel® Dynamic Application Loader (Intel® DAL) basic functionality works properly.
Procedure:	Start test DAL_002 in the Intel® Platform Enablement Test Suite from management console. **Intel® Platform Enablement Test Suite performs the following steps:** 1. Confirm the Intel® Dynamic Application Loader (Intel® DAL) is enabled in firmware. 2. Confirm that the needed Host software components are available (Intel® MEI driver and Intel® Dynamic Application Loader (Intel® DAL) host software). 3. Confirm test application can be installed and loaded to Intel® Dynamic Application Loader. 4. Unload the test application.
Test Pass/Fail Criteria:	All steps return the value "Passed"

Test ID:	**DAL_003**
Test Case Title:	Intel® DAL communication channel exercise
Mandatory/Optional:	Mandatory
Firmware SKU:	Mobile SKU
Description:	Intel® DAL test application is installed and loaded, followed by a communication channel exercise between application and host side application.
Objective:	To test that the Intel® Dynamic Application Loader (Intel® DAL) application can communicate successfully with a host application.
Procedure:	Start test DAL_003 in the Intel® Platform Enablement Test Suite from management console. **Intel® Platform Enablement Test Suite performs the following steps:** 1. Confirm the Intel® Dynamic Application Loader (Intel® DAL) is enabled in firmware. 2. Confirm needed Host software components are available (Intel® MEI driver and Intel® Dynamic Application Loader (Intel® DAL) host software). 3. Exercise basic communication channel between test application and host to verify connectivity flow 4. Unload the test application.
Test Pass/Fail Criteria:	All steps return the value "Passed".

Test ID:	**DAL_004**
Test Case Title:	Intel® DAL Application signing by OEM
Mandatory/Optional:	Mandatory
Firmware SKU:	Mobile SKU
Description:	Intel® DAL test application is installed and loaded, Verifying OEM signing applets by themselves.



Test ID:	**DAL_004**
Objective:	This test is verifying that OEM can sign Intel® DAL applets that were created by the OEM.
Procedure:	1. Confirm the Intel® Dynamic Application Loader (Intel® DAL) is enabled in firmware. 2. Confirm needed Host software components are available (Intel® MEI driver and Intel® Dynamic Application Loader (Intel® DAL) host software). 3. sign Private OEM Key with Security Domain (SD). 4. add SD to the FW. 5. prepare an unsigned Intel DAL applet. 6. sign Intel DAL applet with OEM private key. 7. check applet is running on Platform.
Test Pass/Fail Criteria:	All steps return the value "Passed".

§ §

12 Firmware Capsule Update

Elkhart Lake FW update mechanism supports 'over-the-air firmware update' and local update methods. Firmware components are authenticated before update, and then the platform performs a secure firmware update.

Figure 12-1. Over the Air Firmware Update

Note: All the tests here are per Intel® RVP implementation and not executable on OEM platform. OEMs requested to work with their IBVs to execute their custom capsule update test meeting same objective as test below.

12.1 Test Environment Setup

1. Disable Secure Boot in BIOS setup question to allow firmware upgrade.
2. Boot to BIOS-> Device Manager-> System Setup-> South Cluster Configuration->Miscellaneous Configuration-> BIOS lock ->disable
3. Boot to BIOS-> Device Manager-> System Setup-> South Cluster Configuration-> Miscellaneous Configuration-> Flash Protection Range-> disable
4. Check the system date and time if it is correct or not.
5. Set "testsigning" on through bcdedit using below admin command prompt.
6. "bcdedit /set testsigning on" and reboot the system.
7. Confirm the changes by typing bcdedit in cmd prompt.
8. testsigning Yes

12.2 Tools for Testing

- EHLCapsuleCreate directory
- Keep 32MB IFWI binary and FOTA EFI Driver in Elkhart LakeCapsuleCreate Directory

12.3 Capsule Update Test Coverage Summary

Platform, Operating System Support, How? Column describes the test methodology.

OS Support: W = Microsoft Windows *, AOS = Android OS, U = UEFI shell.

How?: A = Fully Automated using Intel® PETS, I = Interactive using Intel® PETS and M = Manual.

Test ID	Test Case Title	PETS Package Name	OS Supported	How?
CU_001	Create signed capsule image in host system	N/A	W, U	M
CU_002	Update firmware capsule update (Install FW update driver)	N/A	W, U	M
CU_003	Power loss during capsule update with fault tolerance flow	N/A	W, U	M

12.4 Create Signed Capsule Image in Host System

Test ID:	CU_001
Test Case Title:	Create signed capsule image in host system
Objective:	To generate valid capsule binary to be used for capsule update
Test Pass Criteria:	Creation on FW driver package (Capsule update files EHLSystemFw.bin, EHLSystemFw.cat & EHLSystemFw.inf)
Windows* Procedure:	1. Remove Read Only attribute for **EHLCapsuleCreate** directory and its contents. 2. Keep 32MB **IFWI binary** and **FOTA EFI Driver** in **EHLCapsuleCreate** directory. 3. Run **CreateCapsule.bat** in command prompt with below arguments. 4. **CreateCapsule.bat** –b [IFWI binary] –ver [DRIVER VERSION] –arch [Architecture : IA32/X64] 5. Check the "**SysFwCapsule_ARCH_vDRIVER VERSION**" in root directory for three files. (EHLstemFw.bin, EHLsystemfw.cat & EHLSystemFw.inf). ***Note:*** 1. Same IFWI can't be updated on reboot as the Hash will match. If the requirement is at all to update the same version of IFWI, then we need to do clean build for Capsule Update to get processed successfully. 2. The Driver Version is a 4 byte value and must be decimal number, as GenCapsule.exe Utility accepts only decimal value inputs. 3. Clear GPP4 through DnX or tool will cause ESRT entry to get its default values.

12.5 Firmware Capsule Update (Install FW Update Driver)

Test ID:	**CU_002**
Test Case Title:	Update firmware capsule update (Install FW update driver)
Objective:	To Perform a firmware update and verify
Test Pass Criteria:	Updated firmware should have the correct version
Windows* Procedure:	1. Keep the SysFwCapsule_ARCH_vDRIVER VERSION folder in DUT after booting to windows. 2. Do the below things to install the certificate file. a. Double click on the security catalog file (cat file) from package b. Click on View Signature. c. Click on View Certificate followed by clicking on Install Certificate d. For Store Location, choose "Local Machine". Click next and manually browse and select "Trusted Root Certification Authorities". e. Click OK and next to install the certificate on the system for one time. f. You should get the popup message "The Import was Successful" 3. Enter to driver package created. — Run "pnputil -i -a EHLSystemFw.inf" as Administartor, click "Install this driver software anyway". 4. 4.Restart system. Expected Update Capsule Event after restart:- 1. Upon first restart, check the system FW update behavior. During System Firmware update, the message "Please wait while we install a system update" is showed for few seconds followed by reset. 2. System Firmware starts to update for about 1 min with the message "Please wait while we install a system update" at bottom of Screen. After System Firmware update completely, system restarts automatically. 3. Reboot to OS. 4. Check the version of driver in Device Manager->Firmware->EHL System Firmware. — The version of driver is correct. 5. Boot to BIOS setup. Check the versions of IFWI. — The version of IFWI is correct.
EDK Shell Procedure	1. Generate capsule images as described in Section 14.5 2. Copy the generated files (EHLSystemFw.bin, EHLSystemFw.cat & EHLSystemFw.inf) to a USB flash drive. 3. Copy the EHLSystemFw.bin file to eMMC via EFI Shell and rename it to BIOSUpdate.fv 4. Restart the system

12.6 Power Loss during Capsule Update with Fault Tolerance Flow

Test ID:	**CU_003**
Test Case Title:	Stress test while upgrading via capsule by performing G3
Objective:	Interrupt the power source during Capsule Update, power on the system and check if capsule flashing is happening again
Test Pass Criteria:	Capsule Update should happen successfully
Windows* Procedure:	1. Boot to Setup, SUT should boot to OS 2. Set BIOS /BIOS/Device Manager/System Setup/South Cluster Configuration/Miscellaneous Configuration/BIOS Lock = Disable 3. Set BIOS /BIOS/Device Manager/System Setup/South Cluster Configuration/Miscellaneous Configuration/Flash Protect Region Registers = Disable 4. Set BIOS /BIOS/Device Manager/System Setup/Boot/UEFI Security Boot = Disabled 5. Restart the system , SUT should reboot to OS 6. Check device manager for any yellow bangs, SUT should be stable 7. Upgrade IFWI by capsule and interrupt 1 at 75% by Perform G3, IFWI should start getting updated and should get interrupted at 75% by performing G3 8. Press Power button, Reboot the system 9. Wait for 300 seconds, Wait till capsule update continues and boot to OS 10. VERIFY CAPSULE UPDATE **Firmware should get updated**

§ §

13 Intel® Integrated Clock Control Compliancy

This chapter covers details of Intel® ICC test cases supported on Lake Field platforms.

Intel® ICC feature support:

Below table displays ICC Feature/Configuration supported on Elkhart Lake platforms.

ICC Feature/Configuration Supported
• Standard • Adaptive

ICC Profile and parameters configuration recommendation

- Review Intel® Bringup Guide to get familiar with supported frequency and SSC configurations for above features.
- Review Intel ® ICC Tools user guide to get familiar with the ICC SDK
- OEMs are recommended to configure ICC Boot profile and parameters for the profile via Intel® FIT -> ICC tab. Make sure to choose appropriate profile and configure parameters that meet platform and HW requirements.

Intel® PETS test package detail for Intel® ICC

The test cases supported by platforms using Intel® Platform Enablement Test Suite (Intel® PETS) are defined as a part of Compliance_ICC_*.xml.

Note: For Elkhart Lake, for each Intel® ICC test case, the ICC boot profile used by the SUT is checked and only test cases applicable to the currently used boot profile by the SUT are executed. Intel® PETS skips irrelevant tests and does not execute Non applicable test cases.

13.1 Intel® Integrated Clock Control Test Coverage Summary and Details

This section provides summary of ICC test cases applicable to Lake Field based platforms.

Test ID	Test Case Title	Mandatory	PETS/ Manual	Network Factor
ICC_TST_01	Test default settings for Standard configuration	Yes (Only mandatory when SUT's boot profile is selected based on standard profile under FIT or by means of BIOS)	PETS / Manual using ICC SDK embedded	N/A
ICC_TST_02	Test default settings for Adaptive configuration	Yes (Only mandatory when SUT's boot profile is selected based on adaptive profile under FIT or by means of BIOS)	PETS/Manual using ICC SDK embedded	N/A

Test ID	Test Case Title	Mandatory	PETS/ Manual	Network Factor
ICC_TST_04	Test Get and Set of MPHY setting	Yes	PETS/Manual using ICC SDK embedded	N/A

13.2 Intel® Integrated Clock Control Test Cases

13.2.1 Test Default Settings for Standard Configuration

Test ID:	**ICC_TST_01**
Test Case Title:	Test default settings for Standard configuration
Mandatory/Optional:	Mandatory. ***Note:*** Only for SUTs with boot profile that to "standard" profile under FIT ->ICC -> Boot Profile or by means of BIOS ***Note:*** For FIT Tool, Check parameter under FIT| Integrated Clock Controller | Boot Profile selection. if Boot profile selection is based on Standard profile, then this test is mandatory otherwise it can be skipped. ***Note:*** For BIOS, Check parameter using the request to HECI: ICC_GET_PROFILE_REQ if Boot profile selection is based on Standard profile, then this test is mandatory otherwise it can be skipped.
Description:	Verify if the current ICC registers setting in the SUT are set correctly based on standard configuration
Objective:	Ensure that critical ICC register values are configured correctly for standard configuration.
Procedure:	Get BCLK PLL Settings: • API: ICC_GET_CLOCK_SETTINGSEx • library method: *EXTERNAL_API UINT32IccLibGetCurrentClockSettingsWrapper(const ICC_HECI_CLOCK_ID clockId, ICC_GET_CLOCK_SETTINGSEx * const clockSettings);* An error should be returned in case the test has failed
Test Pass/Fail Criteria:	Pass if the critical ICC registers values read are set correctly based on the standard configuration. Frequency= 400 MHZ SSC = 0.5 ***Note:*** For FIT, Check parameter under Flash Image Tool| Integrated Clock Controller | Boot profile selection. If Boot profile is not based on standard profile then this test is expected to fail. ***Note:*** For BIOS, check parameter using the request to HECI : ICC_GET_PROFILE_REQ if Boot profile is not based on standard profile then this test is expected to fail.

13.2.2 Test Default Settings for Adaptive Configuration

Test ID:	**ICC_TST_02**
Test Case Title:	Test default settings for Adaptive configuration
Mandatory/Optional:	Mandatory ***Note:*** Only for SUTs with boot profile set to "Adaptive" profile under FIT ->ICC -> Boot Profile or by means of BIOS. ***Note:*** For FIT Tool, Check parameter under FIT| Integrated Clock Controller | Boot profile selection. if boot profile selection is based on Adaptive profile, This test is mandatory else the user can skip to execute it. ***Note:*** For BIOS check parameter using the request to HECI : ICC_GET_PROFILE_REQ if Boot profile selection is based on Adaptive profile, then this test is mandatory otherwise it can be skipped.
Description:	Verify if the current ICC registers setting in the SUT are set correctly based on Adaptive configuration

Test ID:	**ICC_TST_02**
Objective:	Ensure that critical ICC register values match defaults for Adaptive configuration
Procedure:	Get BCLK PLL Settings: • API: _ICC_SET_CLOCK_SETTINGSEx • Library method: *EXTERNAL_API UINT32IccLibGetCurrentClockSettingsWrapper(const ICC_HECI_CLOCK_ID clockId, ICC_GET_CLOCK_SETTINGSEx * const clockSettings);* An error should be returned in case the test has failed Set the BCLK PLL settings: • API: _ICC_SET_CLOCK_SETTINGSEx • Library method: EXTERNAL_API UINT32 IccLibSetCurrentClockSettingsWrapper(const ICC_HECI_CLOCK_ID clockId, ICC_SET_CLOCK_SETTINGSEx * clockSettings); An error should be returned in case the test has failed
Test Pass/Fail Criteria:	Pass if the critical ICC registers values read are set correctly based on the Adaptive configuration. ***Note:*** For FIT, Check parameter under Flash Image Tool| Integrated Clock Controller | Boot profile selection. If Boot profile is not based on Adaptive profile then this test is expected to fail. ***Note:*** For BIOS check parameter using the request to HECI : ICC_GET_PROFILE_REQ if Boot profile selection is based on Adaptive profile, then this test is mandatory otherwise it can be skipped. ***Note:*** Default frequency and SSC supported for Adaptive is 400MHz with 0.50%. Supported Min.-Max. frequency range is [390.00- 400.00 MHz]. This test checks default configuration for Adaptive clocking. Test may fail if customer change SSC or frequency from default value; however make sure to check if settings are within the expected range supported for Adaptive clocking.

13.2.3 GET and SET MPHY settings

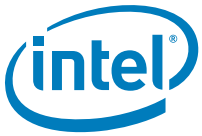
Test ID:	**ICC_TST_04**
Test Case Title:	Get and Set of MPHY setting
Mandatory/Optional:	Mandatory, This is informative test.
Description:	This test output high level detail like CRC count into a bin file , Version and product detail of chipset initialization settings. this test apply a new version of chipset User to manually verify data is correct or not.
Objective:	Verify if correct version of chipset initialization settings are applied or not. In case issue is seen, detail like CRC count, Version and product detail can be used for debug purpose. Apply a new version of chipset initialization settings

Procedure:	GET MPHY Version:
	API:_GET_MPHY_VERSION
	library method: EXTERNAL_API UINT32 IccLibGetMphyVersion(GET_MPHY_VERSION *survTable);
	
	GET MPHY table:
	library method: EXTERNAL_API UINT32 IccLibGetMphySettingsWrapper(UINT32 length, UINT32 offset, UINT8 *buffer,UINT32 *bytesRead);
	
	Set MPHY table:
	library method: EXTERNAL_API UINT32 IccLibSetMphySettingsWrapper(char *mphyFileName);
	
	Note: Retrieving Chipset Initialization file and information can be blocked by some restrictions enforced with End-of-Post being issued. Tester may require to disable End-of-Post message from BIOS menu for the test to successfully pass.
	Note: This test currently displays the command result only.

Test Pass/Fail Criteria:	This is informative test and displays details like CRC count, Version and product detail. User to manually confirm if data looks correct or not.

§ §

< This page is intentionally left blank >



14 Platform Controller Hub (PCH) SoftStrap Configuration

Overview:

The Intel® PCH SoftStraps are load into the appropriate strapping registers within the PCH at boot time from the SPI flash device's Flash Descriptor. Some of the features within the PCH are configurable through the PCH SoftStraps such as the Flexible I/O, SMLINK, GbE, and Intel® ME. The PCH SoftStraps are configure using the FIT tool. Refer to the SPI Programming Guide for the details description on all the available PCH SoftStraps.

All the test case in this chapter are currently cover automatically by PETS on the target system at runtime. Static checking on the image created by FIT is not supported.

Tools for Testing:

Intel® Platform Enablement Test Suite (PETS)—Latest version of tools from this kit. Refer to the Intel® PETS user guide available in the Intel® Compliancy kit for exact instructions on how to load and setup the Intel® PETS software.

Intel® Flash Image Tool (FIT.exe)

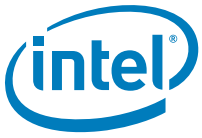
Intel® Flash Programming Tool—Available in DOS (fpt.exe), EFI (fpt.efi), Windows* 32-bit (ftpw.exe), and Windows* 64-bit operating systems.

Test Environment:

The System Under Test (SUT) is to be configured in manual configuration mode a with wired LAN dynamic IP address. The DHCP server connecting the SUT and Management Console (MC) must be configured to ensure that the wired LAN and wireless LAN addresses reside on separate subnets. The MC could be a laptop or desktop system running a version of Windows* supported by PETS. The network configuration consists of a hub or switch, network cables, and a wireless Access Point (AP).

14.1 Test Coverage Summary

Test ID	Test Case Title	PETS/Manual	Network Factor
PSS_003	Flexible I/O Test	PETS	N/A
PSS_004	BIOS Boot-Block Size Test	PETS	N/A

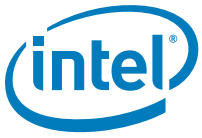


14.2 Flexible I/O Test

Test ID:	**PSS_003**
Test Case Title:	Flexible I/O Test
Mandatory/Optional:	Mandatory
Description:	Flexible I/O is an architecture that allows some high speed signals to be configured as PCIe*, USB 3.x or SATA signals. Through SoftStraps, the functionality on these multiplexed signals are selected to meet I/O needs on the target platform.
Objective:	To verify correct configuration of Flexible I/O SoftStraps.

<table>
<tr><td>Test ID:</td><td>PSS_003</td></tr>
<tr><td>Procedure:</td><td>
Boot to targeted OS. Verify correct configuration of the PCH SoftStraps below:

1. How to get PCIe Controller 1 (Port 0-3) configured?
a. 1x4 – one 4 lane PCIe* Port
<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>PCIe Controller 1 (Port 0-3)</td><td>Offset 0x16D [4:3]</td><td>3h</td></tr>
</table>
i. Are the lanes reversed?
— If Reversed:
<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>PCIe Controller 1 Lane Reversal</td><td>Offset 0x16D [2]</td><td>1h</td></tr>
</table>
— If NOT Reversed:
<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>PCIe Controller 1 Lane Reversal</td><td>Offset 0x16D [2]</td><td>0h</td></tr>
</table>
b. 2x2 – two 2 lane PCIe* Port
<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>PCIe Controller 1 (Port 0-3)</td><td>Offset 0x16D [4:3]</td><td>2h</td></tr>
</table>
c. 1x2, 2x1- One 2 lane PCIe* Port, Two 1 lane PCIe* Port
<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>PCIe Controller 1 (Port 0-3)</td><td>Offset 0x16D [4:3]</td><td>1h</td></tr>
</table>
d. 4x1: Ports (1-4) (x1)
<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>PCIe Controller 1 (Port 0-3)</td><td>Offset 0x16D [4:3]</td><td>0h</td></tr>
</table>
2. How do you have PCIe Multi VC Controller 1 (Ports 7-11) configured?
a. 1x1 PCIe* Port on lane 7.
<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>PCIe Multi VC Controller 1 (Port 7-11)</td><td>Offset 0x174 [3:1]</td><td>0h</td></tr>
<tr><td>FIA_LOSL7 (FIA/LOSL7)</td><td>Offset 0x1CC [3:0]</td><td>Bh</td></tr>
</table>
b. 1x1 PCIe* Port on lane 8.
<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>PCIe Multi VC Controller 1 (Port 7-11)</td><td>Offset 0x174 [3:1]</td><td>0h</td></tr>
<tr><td>FIA_LOSL8 (FIA/LOSL8)</td><td>Offset 0x1CC [7:4]</td><td>Bh</td></tr>
</table>
c. 1x1 PCIe* Port on lane 10.
<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>PCIe Multi VC Controller 1 (Port 7-11)</td><td>Offset 0x174 [3:1]</td><td>0h</td></tr>
<tr><td>FIA_LOSL10 (FIA/LOSL10)</td><td>Offset 0x1CD [7:4]</td><td>Bh</td></tr>
</table>
</td></tr>
</table>



Test ID:	PSS_003

d. 1x2 PCIe Port on Lanes 7 & 9

Name	Location	Value
PCIe Multi VC Controller 1 (Port 7-11)	Offset 0x174 [3:1]	1h
FIA_LOSL7 (FIA/LOSL7)	Offset 0x1CC [3:0]	Bh
FIA_LOSL9 (FIA/LOSL9)	Offset 0x1CD [3:0]	Bh

e. 1x2 PCIe Port on Lanes 8 & 9

Name	Location	Value
PCIe Multi VC Controller 1 (Port 7-11)	Offset 0x174 [3:1]	1h
FIA_LOSL8 (FIA/LOSL8)	Offset 0x1CC [7:4]	Bh
FIA_LOSL9 (FIA/LOSL9)	Offset 0x1CD [3:0]	Bh

f. 1x2 PCIe Port on Lanes 10 & 11

Name	Location	Value
PCIe Multi VC Controller 1 (Port 7-11)	Offset 0x174 [3:1]	1h
FIA_LOSL10 (FIA/LOSL10)	Offset 0x1CD [7:4]	Bh
FIA_LOSL11 (FIA/LOSL11)	Offset 0x1CE [3:0]	Bh

3. How do you have PCIe Multi VC Controller 2 (Ports 2-5) configured?

a. 1x1 PCIe* Port on lane 2.

Name	Location	Value
PCIe Multi VC Controller 2 (Port 2-5)	Offset 0x178 [3:1]	0h
FIA_LOSL2 (FIA/LOSL2)	Offset 0x1C9 [3:0]	Bh

b. 1x1 PCIe* Port on lane 4.

Name	Location	Value
PCIe Multi VC Controller 2 (Port 2-5)	Offset 0x178 [3:1]	0h
FIA_LOSL4 (FIA/LOSL4)	Offset 0x1CA [7:4]	Bh

c. 1x2 PCIe Port on Lanes 2 and 3

Name	Location	Value
PCIe Multi VC Controller 2 (Port 2-5)	Offset 0x178 [3:1]	1h
FIA_LOSL2 (FIA/LOSL2)	Offset 0x1C9 [3:0]	Bh
FIA_LOSL3 (FIA/LOSL3)	Offset 0x1CA [3:0]	Bh

d. 1x2 PCIe Port on Lanes 4 and 5

Name	Location	Value
PCIe Multi VC Controller 2 (Port 2-5)	Offset 0x178 [3:1]	1h
FIA_LOSL4 (FIA/LOSL4)	Offset 0x1CA [7:4]	Bh
FIA_LOSL5 (FIA/LOSL5)	Offset 0x1CB [3:0]	Bh

4. How do you have PCIe Multi VC Controller 3 (Ports 6-7) configured?

a. 1x1 PCIe* Port on lane 6.

Name	Location	Value
PCIe Multi VC Controller 3 (Port 6-7)	Offset 0x17C [3:1]	0h
FIA_LOSL6 (FIA/LOSL6)	Offset 0x1CB [7:4]	Bh

Test ID:	PSS_003

b. 1x2 PCIe* Port on lanes 6 & 7.

Name	Location	Value
PCIe Multi VC Controller 3 (Port 6-7)	Offset 0x17C [3:1]	0h
FIA_LOSL6 (FIA/LOSL6)	Offset 0x1CB [7:4]	Bh
FIA_LOSL7 (FIA/LOSL7)	Offset 0x1CC [3:0]	Bh

5. Does this platform use PCH PCIe port 3 as USB3 Port 3?

— If yes, PCH PCIe Port 3 configured as USB3

Name	Location	Value
USB3 / PCIe Combo Port 0 (FIA/LOSL2)	Offset 0x1C9 [7:4]	1h

— If no, PCH PCIe Port 3 configured as PCIe

Name	Location	Value
USB3 / PCIe Combo Port 0 (FIA/LOSL2)	Offset 0x1C9 [7:4]	5h

6. Does this platform use PCH PCIe Port 4 as USB3 Port 4?

— If yes, PCH PCIe Port 4 configured as USB3

Name	Location	Value
USB3 / PCIe Combo Port 1 (FIA/LOSL3)	Offset 0x1CA [3:0]	1h

— If no, PCH PCIe Port 4 configured as PCIe

Name	Location	Value
USB3 / PCIe Combo Port 1 (FIA/LOSL3)	Offset 0x1CA [3:0]	5h

1. How is SATA / PCIe* Combo Port 0 Strap configured on the platform?

i. Statically assigned to SATA Port 0.

Name	Location	Value
SATA / PCIe Combo Port 0 (FIA/LOSL10)	Offset 1CD [7:4]	7h

ii. Statically assigned to PCIe* Port 10.

Name	Location	Value
SATA / PCIe Combo Port 0 (FIA/LOSL10)	Offset 1CD [7:4]	5h

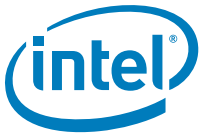
2. How is SATA / PCIe* Combo Port 1 Strap configured on the platform?

i. Statically assigned to SATA Port 1.

Name	Location	Value
SATA / PCIe Combo Port 1 (FIA/LOSL11)	Offset 1CE [3:0]	7h

ii. Statically assigned to PCIe* Port 11.

Name	Location	Value
SATA / PCIe Combo Port 1 (FIA/LOSL11)	Offset 1CE [3:0]	5h



14.3 BIOS Boot-Block Size Test

<table>
<tr><td>Test ID:</td><td>PSS_004</td></tr>
<tr><td>Test Case Title:</td><td>BIOS Boot-Block size Test</td></tr>
<tr><td>Mandatory/Optional:</td><td>Mandatory (SPI Configurations Only)</td></tr>
<tr><td>Description:</td><td>BIOS Boot-Block size deals with a BIOS recovery mechanism. If this is not set correctly, then BIOS boot-block recovery mechanism will not work.</td></tr>
<tr><td>Objective:</td><td>To verify BIOS boot-block size of correctly setup.</td></tr>
<tr><td>Procedure:</td><td>
Boot to targeted OS. Verify correct configuration of the PCH SoftStraps below:

1. What size is SPI flash BIOS boot block?

a. If 64KB

<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>Top Swap Block size</td><td>Offset 0x114 [6:4]</td><td>0h</td></tr>
</table>

b. 128KB

<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>Top Swap Block size</td><td>Offset 0x114 [6:4]</td><td>1h</td></tr>
</table>

c. 256KB

<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>Top Swap Block size</td><td>Offset 0x114 [6:4] LP and H</td><td>2h</td></tr>
</table>

d. 512KB

<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>Top Swap Block size</td><td>Offset 0x114 [6:4] LP and H</td><td>3h</td></tr>
</table>

e. 1MB

<table>
<tr><th>Name</th><th>Location</th><th>Value</th></tr>
<tr><td>Top Swap Block size</td><td>Offset 0x114 [6:4] LP and H</td><td>4h</td></tr>
</table>
</td></tr>
<tr><td>Test Pass/Fail Criteria:</td><td>Test passes if SoftStraps/register setting in this step matches to the configuration in the target system.</td></tr>
</table>

§ §