DFx Secure Plugin

UNIT VERIFICATION PLAN REFERENCE

IP Rev. PIC5_RTL1P0_V3
January 2019

Intel Restricted Secret

This document contains information on products in the design phase of development.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED OR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked “reserved” or “undefined.” Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document 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.

Contact your Intel account manager or distributor to obtain the latest specifications and before placing your product order.

Copies of documents that have an order number and are referenced in this document or in other Intel literature can be obtained from your Intel account manager or distributor.

Contents

1 About This Document

1.1 How to Use This Template

Please do not remove any headings from this document. If you do not need the headings to describe your IP, enter "Not applicable" under the heading. This lets the reader know that we did not overlook this topic.

In the main document that follows, add new headings that you need to fully describe the architecture of this IP. (Please add them in the appropriate chapters.)

Note: Most red text in this document contains instructions for filling out the section where it appears. The tag for most of this red text is called "Gaps." You should replace this text with the content appropriate for that section, ensuring that the text is tagged appropriately (for example, with the BodyText or List Bullet style). If a section is not relevant, do not remove it, but just replace the Gap text with "Not applicable" and apply the BodyText style.

1.2 Goal of This Document

This document contains all information a verification team needs to accomplish the IP verification task without needing to seek help from another source.

1.3 Audience

The information in this document is intended to describe the verification strategy and execution plans for this IP. It is written as a Test Plan for the IP verification team, describing how the IP *is going to be* tested. The same document is published as a Verification Reference for a SoC team to see how the IP *was* tested.

1.4 Supported Projects

This document supports the following projects at the listed RTL maturity level.

Table 1. RTL Support Level for SoC Projects

Project Name	IP Maturity Level
ALL	PIC4

1.5 Document Revision History

Table 2. Revision History of This Document

Author	Revision No.	Description	Revision Date
Bulusu, Shivaprashant U, Bharath	Rev1.0	Final version of Verification reference	01/04/2014
Bandana, V sudheer	Rev1.0	Verification Reference updated with HDK.	08/04/2015

1.6 Contact Information

Table 3. Contact Information for the XX IP

Name	Function	Email
Bulusu, Shivaprashant	Verification	shivaprashant.bulusu@intel.com
Bandana, V, Sudheer	Verification	sudheer.v.bandana@intel.com
Adithya, BS	Design	b.s.adithya@intel.com
Susann Flowers	Doc Template Owner	susann.flowers@intel.com

1.7 Related Documents

If you need more information on this IP, you may find these documents helpful:

- SIP_DfxSecurePlugin_productBrief
- SIP_DfxSecurePlugin_integrationGuide
- IOSF_DFx_HAS_rev1.2

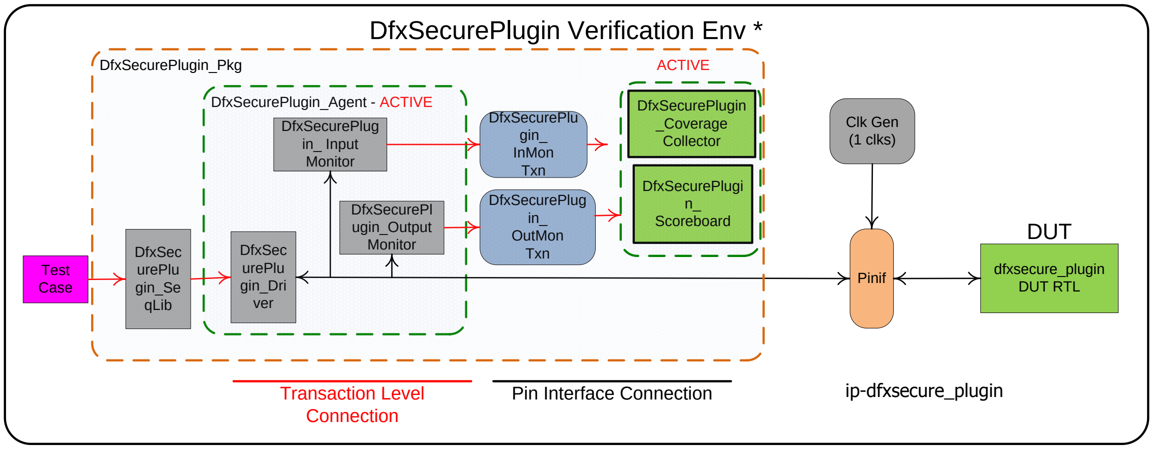
2 Overview

2.1 IP Description

The DFx secure plugin IP contains a group of signals composed of one bus and two control signals that distributes a uniform policy value to all IP-blocks in the SoC. The signals are defined as fdfx_earlyboot_exit, fdfx_secure_policy [3:0] and fdfx_policy_update. IP-blocks instantiate a security plug-in IP that translates the current policy and enables or disables access to DFx features within the agent or IP-block. The DFx secure policy signal group is composed of a binary encoded policy bus, a latch enable and an early boot debug exit signals. This signal group is distributed from a centralized DFx security aggregator.

2.2 Testbench Description

The Verification ENV is in OVM methodology. The Testbench architecture is illustrated in the figure below. The exact files for each block are provided in the diagram. The Env class is extended from the `ovm_env`. It instantiates all the OVM components: master agent, slave agent, Scoreboard, Coverage, and OVM report.


DfxSecurePlugin Verification Env *
DfxSecurePlugin_Pkg
ACTIVE
DfxSecurePlugin_Agent - ACTIVE
DfxSecurePlugin_ Input Monitor
DfxSecurePlugin_ InMon Txn
DfxSecurePlugin _Coverage Collector
DfxSecurePlugin_ Scoreboard
DfxSecurePlugin_Output Monitor
DfxSecurePlugin_ OutMon Txn
Clk Gen (1 clks)
Test Case
DfxSecurePlugin_SeqLib
DfxSecurePlugin_Driver
Pinif
DUT
dfxsecure_plugin DUT RTL
Transaction Level Connection
Pin Interface Connection
ip-dfxsecure_plugin

2.3 Verification Scope

All features of this IP are verified.

2.4 Dependencies/Assumptions

Not applicable to this IP

3 Verification Strategy

3.1 High-Level Verification Strategy

3.1.1 Methodology

The Verification ENV is in OVM methodology.

3.1.2 Stimulus Strategy

The driver models the behavior of the Security Aggregator block in an SoC.

3.1.3 Coverage Strategy

Functional coverage is coded to see if all the scenarioes are covered. Code coverage is also analyzed to see if all the toggle, branch, conditions and fsm state transitions are getting covered.

3.1.4 Checking Strategy

This IP uses assertion based checking.

3.1.5 Formal Verification Strategy

Not applicable to this IP

3.1.6 Debug Strategy

Not applicable to this IP

3.1.7 Security Strategy

This IP name itself says that it is a IP which provides Security to other IP where ever it is used. So this whole document will say about the Verification of this IP.

3.2 Verification Strategy for Areas of Special Emphasis

3.2.1 Reset Verification

Not applicable to this IP

3.2.2 Control Register and Fuse Verification

Not applicable to this IP

3.2.3 Power Management Verification

Not applicable to this IP

3.2.4 Mixed Signal Verification

Not applicable to this IP

3.2.5 Performance Verification

Not applicable to this IP

3.2.6 Security Verification

Not applicable to this IP

3.2.7 Error Scenario Verification

Not applicable to this IP

3.2.8 Design for Test (DFT) Verification

Not applicable to this IP

3.2.9 Design for Validation (DFV) Verification

Not applicable to this IP

3.2.10 Firmware Verification

Not applicable to this IP

3.2.11 Software / Driver Verification

Not applicable to this IP

3.2.12 Timer / Counter Verification

There are no timers and counters.

3.3 Reuse Strategy

DFx Secure Plugin BFM is used to drive the interface of this IP.

3.3.1 Local Reuse for IP Verification

Not applicable to this IP

3.3.2 Reuse of IP Verification collateral for SoC

There are no requirements for this IP’s verification Env to be used at integration level.

4 Testbench Details

4.1 Testbench Tools

4.2 Environment Setup and Test Run

- Navigate to the directory where you downloaded the IP package.
- To compile the model:

```
make run_vcs
```

- To run a sample IP-level DfxSecurePlugin_PolicySweepSeqTest test (included in the release):

```
   simbuild -dut dfxsecure_plugin -ace_args ace -x -t
DfxSecurePlugin_PolicySweepSeqTest -simv_args "+VPD" -ace_args- -1c -CUST
<CUST> -1c-
   simbuild -dut dfxsecure_plugin -ace_args ace -x -t
DfxSecurePlugin_PolicySweepSeqTest -simv_args "+FSDB" -ace_args- -1c -CUST
<CUST> -1c-
```

4.3 Simulation Stages

Not applicable to this IP

4.4 Testbench Monitors

Component	Description
DfxSecurePlugin_InpMonitor.sv	Monitors the input pins going to DUT
DfxSecurePlugin_OutMonitor.sv	Monitors the output pins going to DUT
DfxSecurePlugin_OutMonTxn.sv	Contains the packet that to be sent to Scoreboard from the Output Monitor
DfxSecurePlugin_InpMonTxn.sv	Contains the packet that to be sent to Scoreboard from the Input Monitor.

4.5 Testbench Output

The testcase run creates logfiles in the following directory:

$IP_ROOT/pwa/results/tests/<TESTCASE_NAME>/

The following log files and tracker files will be created.

- <TESTCASE_NAME.log>
- postsim.log

4.6 Testbench Directory Structure

```
`-- verif
    |-- lib
    |   `-- ovm_pkg.hdl
    |-- tb
```

```
|   |-- DfxSecurePlugin_Agent.sv
|   |-- DfxSecurePlugin_ClkGen.sv
|   |-- DfxSecurePlugin_Coverage.sv
|   |-- DfxSecurePlugin_Driver.sv
|   |-- DfxSecurePlugin_Env.sv
|   |-- DfxSecurePlugin_InpMonTxn.sv
|   |-- DfxSecurePlugin_InpMonitor.sv
|   |-- DfxSecurePlugin_OutMonTxn.sv
|   |-- DfxSecurePlugin_OutMonitor.sv
|   |-- DfxSecurePlugin_Pkg.hdl
|   |-- DfxSecurePlugin_Pkg.sv
|   |-- DfxSecurePlugin_Scoreboard.sv
|   |-- DfxSecurePlugin_SeqDrvTxn.sv
|   |-- DfxSecurePlugin_SeqLib.sv
|   |-- DfxSecurePlugin_Seqr.sv
|   |-- DfxSecurePlugin_TbDefines.svh
|   |-- DfxSecurePlugin_Tbtop.sv
|   |-- DfxSecurePlugin_TestIsland.sv
|   |-- DfxSecurePlugin_VifContainer.sv
|   |-- DfxSecurePlugin_pin_if.sv
|   |-- DfxSecurePlugin_tb.hdl
|   |-- DfxSecurePlugin_ti.hdl
|   `-- include
`-- tests
    |-- DfxSecurePlugin_BaseTest.sv
    |-- DfxSecurePlugin_Test.sv
    |-- DfxSecurePlugin_TestLib.sv
    `-- DfxSecurePlugin_tests.hdl
```

4.7 Testbench Utilities for Address, IP State and Memory

Not applicable to this IP

5 Stimulus Details

5.1 IP Power Up and Reset

There is only one reset in the design: fdfx_powergood

There is no minimum assertion time for this reset. The latch in the design is cleared by fdfx_powergood. This reset is similar to powergood_rst_b in the SoC, an active low signal. Features that are enabled by security can work only after `early_boot_exit` goes high.

5.2 Initial IP Configuration

Not applicable to this IP

5.3 Dynamic IP Configuration

Not applicable to this IP

5.4 Dynamic Injectors

Not applicable to this IP

5.5 Stimulus Generation

Not applicable to this IP

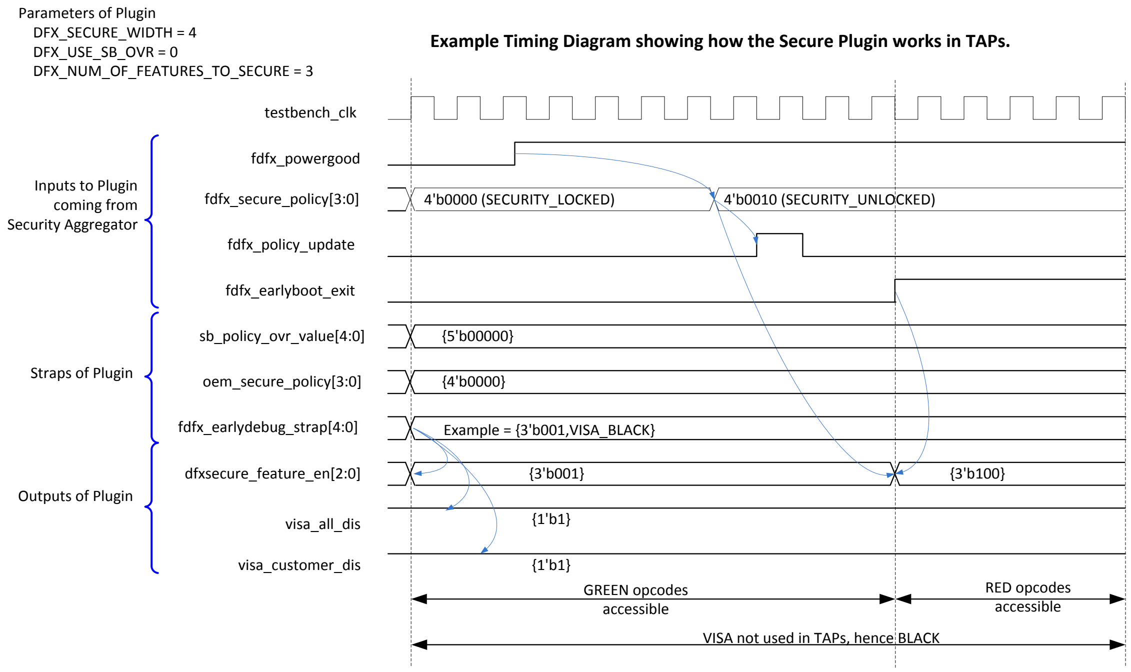
5.6 Transaction Classes / Sequence Items

Transaction Class	Description
DfxSecurePlugin_SeqDrvTxn	This is the Packet between the Sequencer and the Driver.

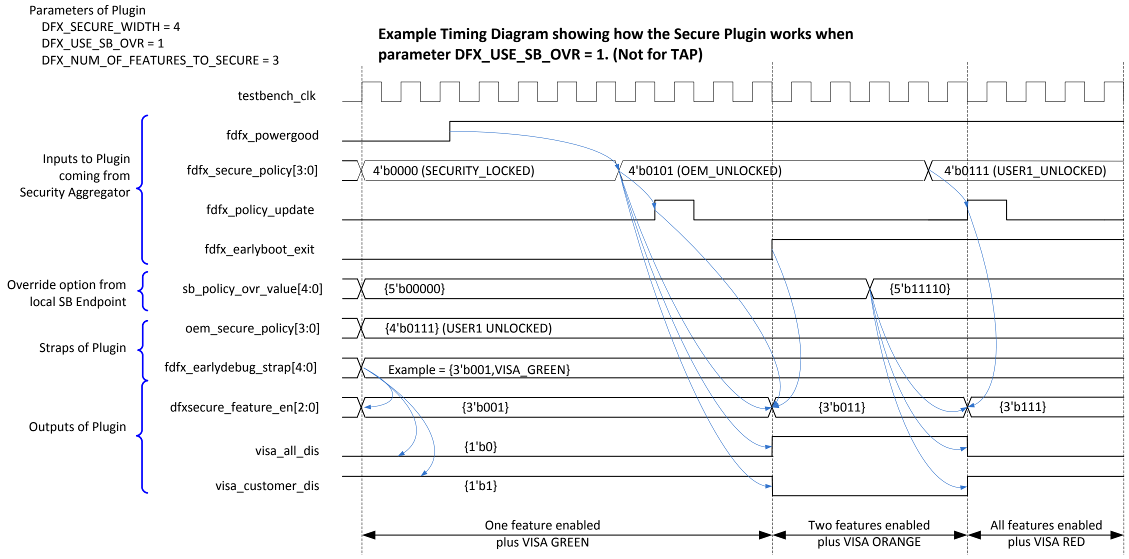
5.7 Sequencers / Sequence Drivers

The driver models the behavior of the Security Aggregator block in an SoC. It drives the signals as shown in the next two diagrams.

Driver Timing (DFX_USE_SB_OVR = 0)



Driver Timing (DFX_USE_SB_OVR = 1)


Parameters of Plugin
DFX_SECURE_WIDTH = 4
DFX_USE_SB_OVR = 1
DFX_NUM_OF_FEATURES_TO_SECURE = 3
Example Timing Diagram showing how the Secure Plugin works when parameter DFX_USE_SB_OVR = 1. (Not for TAP)
testbench_clk
fdfx_powergood
Inputs to Plugin coming from Security Aggregator
fdfx_secure_policy[3:0]
4'b0000 (SECURITY_LOCKED)
4'b0101 (OEM_UNLOCKED)
4'b0111 (USER1_UNLOCKED)
fdfx_policy_update
fdfx_earlyboot_exit
Override option from local SB Endpoint
sb_policy_ovr_value[4:0]
{5'b00000}
{5'b11110}
Straps of Plugin
oem_secure_policy[3:0]
{4'b0111} (USER1 UNLOCKED)
fdfx_earlydebug_strap[4:0]
Example = {3'b001,VISA_GREEN}
Outputs of Plugin
dfxsecure_feature_en[2:0]
{3'b001}
{3'b011}
{3'b111}
visa_all_dis
{1'b0}
visa_customer_dis
{1'b1}
One feature enabled plus VISA GREEN
Two features enabled plus VISA ORANGE
All features enabled plus VISA RED

5.8 Sequences / Sequence Libraries

Test Sequence Name	Parameters	Function

DfxSecurePlugin_BaseSeq		Places the DUT in default mode, drives the fdfx_powergood reset asynchronously and drives the signals based on DFX_USE_SB_OVR.
		When DFX_USE_SB_OVR =0
		fdfx_secure_policy = 'b0
		sb_policy_ovr_value = 'b0
		oem_secure_policy = 'b0
		When DFX_USE_SB_OVR =1
		fdfx_secure_policy = 'b0
		sb_policy_ovr_value = random value (where the range depends on DFX_NUM_OF_FEATURES_TO_SECURE)
		oem_secure_policy = 0 to 15

DfxSecurePlugin_DrivePowerGoodSeq		Issues a low to high transition on the fdfx_powergood signal.
DfxSecurePlugin_PolicySweepSeq	DFX_SECURE_POLICY_MATRIX	Places the DUT in DfxSecurePlugin_PolicySweepSeq mode. Drives the fdfx_powergood reset asynchronously and sweeps all 16 policies over fdfx_secure_policy irrespective of DFX_USE_SB_OVR. The driving of sb_policy_ovr_value and oem_secure_policy depends on DFX_USE_SB_OVR as follows: When DFX_USE_SB_OVR =0 sb_policy_ovr_value = 'b0 oem_secure_policy = 'b0 When DFX_USE_SB_OVR =1 sb_policy_ovr_value = random value (where the range depends on DFX_NUM_OF_FEATURES_TO_SECURE) oem_secure_policy = 0 to 15

Test Sequence Name	Parameters	Function
DfxSecurePlugin_DriveUserPolicySeq	DFX_SECURE_POLICY_MATRIX	Places the DUT in DfxSecurePlugin_DriveUserPolicySeq mode. Drives the user defined policies over fdfx_secure_policy (ranging from 0 to 15). The driving of sb_policy_ovr_value and oem_secure_policy depends on DFX_USE_SB_OVR as follows: When DFX_USE_SB_OVR =0 sb_policy_ovr_value = 'b0 oem_secure_policy = 'b0 When DFX_USE_SB_OVR =1 sb_policy_ovr_value = random value (where the range depends on DFX_NUM_OF_FEATURES_TO_SECURE) oem_secure_policy =Strap Value.

DfxSecurePlugin_DriveUserPolicyViaSBOVRSeq		Places the DUT in DfxSecurePlugin_DriveUserPolicyViaSBOVRSeq mode and drives the signals based on DFX_USE_SB_OVR as follows:
		When DFX_USE_SB_OVR =0
		fdfx_secure_policy = 'b0
		sb_policy_ovr_value = 'b0
		oem_secure_policy = 'b0
		When DFX_USE_SB_OVR =1
		fdfx_secure_policy = 'b0
		sb_policy_ovr_value = random value (where the range depends on DFX_NUM_OF_FEATURES_TO_SECURE)
		oem_secure_policy = 0 to 15

5.9 Tests and Test Templates

Directed Test

Name	Description	MS	Status
DfxSecurePlugin_DefaultTest	This test places the DUT in Default mode.	1.0	Passing
DfxSecurePlugin_DriveUserPolicySeqTest	This test place the DUT in test_driver mode. Drive the reset asynchrously. Drive all 16 policies in DFX_SECURE_POLICY_MATRIX	1.0	Passing
DfxSecurePlugin_DriveAllUserInputSeqTest	This test place the DUT in test_driver mode. Drive User defined inputs to DUT	1.0	Passing
DfxSecurePlugin_PolicySweepSeqTest	This test place the DUT in drive_user_policy mode. Drive the reset asynchrously. Drive the user defined policies on DFX_SECURE_POLICY_MATRIX	1.0	Passing

5.10 Test Lists and Regressions

Name	Description	MS	Status
run_all_configs	This script runs the regression for all parameter combinations. This script is located in $IP_ROOT/scripts	1.0	Coded

6 Coverage Details

6.1 Coverage Tools

VCS is used to collect the coverage.

6.2 Functional Coverage Conditions

Functional Coverage

Group	Name	Cases	Description	MS	Status
cg1_control	fdfxpowergood	2	Covers powergood for high and low value		Coded
cg1_control	fdfxearlbootexit	2	Covers earlyboot_exit for high and low value		Coded
cg1_control	visaalldis	2	Covers visa_all_dis for high and low value		Coded
cg1_control	visacustomerdis	2	Covers visa_customer_dis for high and low value		Coded
cg1_control	crossvisas	2	Cross between visa_all_dis and visa_customer_dis		Coded
cg2_policy_use1	fdfxsecurepolicy	16	DFX_USE_SB_OVR = 1 Covers various values of security policy		Coded
cg2_policy_use1	sbpolicyovrvalue	16	DFX_USE_SB_OVR = 1 Cover various values of sideband policy override values		Coded
cg2_policy_use1	oemsecurepolicy	8	DFX_USE_SB_OVR = 1 Cover various values of oem_secure_policy		Coded
cg2_policy_use1	dfxsecurefeatureen	8	DFX_USE_SB_OVR = 1 Cover various values of dfxsecurefeatureen		Coded
cg3_policy_use0	fdfxsecurepolicy	16	DFX_USE_SB_OVR = 0 Covers various values of security policy		Coded
cg3_policy_use0	sbpolicyovrvalue	0	DFX_USE_SB_OVR = 0 Cover various values of sideband policy override values		Coded
cg3_policy_use0	oemsecurepolicy	0	DFX_USE_SB_OVR = 0 Cover various values of oem_secure_policy		Coded
cg3_policy_use0	dfxsecurefeatureen	8	DFX_USE_SB_OVR = 0 Cover various values of dfxsecurefeatureen		Coded

6.3 Code Coverage

Code coverage analysis is done. Coverage includes condition, line, toggle, branch, fsm, assert.

6.4 Coverage Indicators

Scripts to run Coverage:

For Coverage_Use0,

```
simbuild -dut dfxsecure_plugin -ace_args source scripts/get_coverage_use0 -
ace_args- -1c -CUST <CUST> -1c-
```

For Coverage_Use1,

```
simbuild -dut dfxsecure_plugin -ace_args source scripts/get_coverage_use1 -
ace_args- -1c -CUST <CUST> -1c-
```

For Coverage_use0

SCORE	LINE	COND	TOGGLE	FSM	BRANCH	ASSERT	GROUP
100%	100%	100%	100%	100%	100%	100%	100%

For Coverage_use1

SCORE	LINE	COND	TOGGLE	FSM	BRANCH	ASSERT	GROUP
100%	100%	100%	100%	100%	100%	100%	100%

For Use0, See exlcusion file in path: $IP_ROOT/verif/tb/include/dsp_cov_exclusions.el

For Use1, See exlcusion file in path: $IP_ROOT/verif/tb/include/dsp_cov_exclusions_use1.el

7 Checking Details

Embedded SVAs are used throughout the design to detect illegal conditions.

Name	Description	MS	Status
CHK_DFX_SECURE_WIDTH_IS_FOUR	To check DFX_SECURE_WIDTH parameter value	1.0	Complete
CHK_SB_POLICY_OVR_VALUE_IS_ZERO_IF_USE_OVR_IS_ZERO	To check sb_policy_ovr_value input is equal to 0x0 when DFX_USE_SB_OVR is equal to 0	1.0	Complete
CHK_OEM_SECURE_POLICY_IS_ZERO_IF_USE_OVR_IS_ZERO:	To check oem_secure_policy input value equal to 0x0 when DFX_USE_SB_OVR is equal to 0	1.0	Complete
CHK_OEM_SECURE_POLICY_IS_USER_DEFINED_POLICY_STATES_IF_USE_OVR_IS_HIGH	To check oem_secure_policy is set to one of the user defined (unused) policy states when the parameter USE_SB_OVR is set to logic 1. The user defined policy states are user 1 through 8 unlocked (hex values 0x7 through 0xE).	1.0	Complete
chk_check_output_eq_sb_ovr_val_when_use_sb_enabled_and_oem_eq_policy:	To check DFxSecure Feature En & Visa Feature En values equal to sb_policy_ovr_value when DFX_USE_SB_OVR equal to 1 and dfxsecure_feature_lch equal to oem_secure_policy value;	1.0	Complete
chk_check_output_eq_plcy_matrix_val_when_use_sb_enabled_and_oem_not_eq_policy:	To check DFxSecure Feature En & Visa Feature En values equal to dfxsecure_feature_int when DFX_USE_SB_OVR equal to 1 and dfsecure_feature_lch not equal to oem_secure_policy value;	1.0	Complete
chk_check_output_eq_plcy_matrix_val_when_use_sb_disabled_and_early_bootexit_high:	To check DFxSecure Feature En & Visa Feature En values equal to dfxsecure_feature_int when DFX_USE_SB_OVR equal to 0 and fdfx_earlyboot_exit equal to 1;	1.0	Complete
chk_check_output_eq_early_bootexit_val_when_early_bootexit_low:	To check DFxSecure Feature En & Visa Feature En values equal to fdfx_earlydebug_strap when DFX_USE_SB_OVR equal to 0 and fdfx_earlyboot_exit equal to 0	1.0	Complete

8 Debug Details

Not applicable to this IP

9 Formal Verification Details—OPTIONAL if no FV/FPV

Not applicable to this IP

10 IP Verification Milestones

10.1 Milestones

Table 1 : IP Development Verification Milestones

Milestone	Verification Tasks	Status
IP Val 1.0	(Prerequisite: HAS 1.0)	Received
	Overall regression pass rate of 100% with no failures	Complete
	Coverage at 100%	Complete

10.2 Other Indicators

Not applicable to this IP

11 Glossary

This chapter defines key terms used in this document.

Term	Definition
Terms	
DSP	Dfx-SecurePlugin
Acronyms and Abbreviations	
STAP	Slave TAP