cadence

Cadence Driver Verification Plan and Test Report

Product Version 1.0.3

January 2024

Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

Cadence Driver Verification Plan and Test Report

Cadence Design Systems

Hardware Identifier:3b7280f2c9ba6578cc418ade334264cc

Table of Contents

List of Figures

List of Tables

Chapter 1. Overview

1.1. Document Purpose

This document explains how the core driver was verified and presents the test results and source code analysis.

- The verification environment is described in chapter 2.
- The test results are presented in chapters 3.
- The result of source code analysis starts from chapter 4.

1.2. Acronyms

The following table lists abbreviations that are commonly used in this document.

Core Driver	Cadence Firmware component that provides IP programming abstraction.
CPU	Central Processing Unit
RTL	Register Transfer Level

Chapter 2. Test Equipment/Environment

2.1. Overview

All core driver testing is on a hardware simulated platform, running a bare-metal build against RTL IP. For verification purposes, whole computer systems (processor, controller under tests, interrupts controller, memory) are simulated using Cadence's simulator. As part of the testing, device under tests are in wrapper. This allows the system to verify, that the controller works properly. A sequence of tests is executed within the VSP environment. Each test is marked as finished properly when test criteria were met. For testing two types of scenarios were used:

Positive For this type of tests, device under test perform an operation. Operation should compete successfully.

Negative For this type of tests, the required operation will not be performed, and driver shall return an error.

The system uses error injection when testing negative scenarios. For positive scenarios regular version of CPS is sufficient.

2.2. RTL Environment

The simulated hardware environment utilizes Cadence Firmware Verification Platform to connect the driver to the actual RTL for your IP Controller. The architecture is as follows:

Figure 2.1. Architecture of RTL environment.

2.3. Verification environment

Verification environment used for tests is built on RedHat 6.5 64-bit edition. For testing purposes such modules were used:

2.3.1. license

```
This module is loaded
```

2.3.2. incisive/152/15.20.037

```
set             cds_root          /grid/avs/install
set             cds_pkg           incisive/15.2/15.20.037
setenv   INCISIVE_HOME $cds_root/$cds_pkg
setenv   AMS_HOME  $cds_root/$cds_pkg
append-path     PATH  $cds_root/$cds_pkg/bin
append-path     PATH  $cds_root/$cds_pkg/tools/bin
append-path     PATH  $cds_root/$cds_pkg/tools/systemc/gcc/bin
```

2.3.3. uxe/171/17.1.0.p48

```
set             cds_root          /grid/scp/hw/tools
set             cds_pkg           UXE171/17.1.0.p48
setenv          AXIS_HOME         $cds_root/$cds_pkg/tools
append-path     PATH              $cds_root/$cds_pkg/tools/bin
```

2.3.4. mmp/152/15.2.0

```
set             cds_root          /grid/scp/hw/tools
set             cds_pkg           MMP152/15.2.0
setenv          MMP_HOME          $cds_root/$cds_pkg
```

2.3.5. vipcat/113/11.30.052

```
set             log_usage       "/home/scpadmin/bin/logModuleUsage.csh"
set             cds_pkgname     [module-info name]
set             cds_pkgver      [lindex [split $cds_pkgname "/" ] end]
set             cds_pkg         ${cds_pkgver}-s
set cds_root ""
prepend-path    SPECMAN_PATH    $cds_root/$cds_pkg/packages
set             denaliBase      $cds_root/$cds_pkg/tools.lnx86/denali
set             denaliBase64    $cds_root/$cds_pkg/tools.lnx86/denali_64bit
setenv          DENALI          $denaliBase
append-path     PATH            $denaliBase/bin
append-path     PATH            $cds_root/$cds_pkg/bin
append-path     LD_LIBRARY_PATH "$denaliBase/lib"
append-path     LD_LIBRARY_PATH "$denaliBase64/lib"
```

2.3.6. xtensa/RF-2016.4

```
set  cds_root          /grid/scp/hw/tools
set   cds_pkg           xtensa/RF-2016.4
set  sw_tools   $cds_root/$cds_pkg/XtDevTools/install/tools/RF-2016.4-linux/XtensaTools
setenv  XTENSA_SW_TOOLS   $sw_tools
setenv  XTENSA_SYSTEM   $sw_tools/config
append-path PATH           $cds_root/$cds_pkg/Xplorer-6.0.4
append-path PATH           $sw_tools/bin
```

2.3.7. arm-none-eabi/8.3.0

```
append-path PATH /grid/scp/hw/tools/arm-none-eabi/8.3.0/armv7a-none-eabi/bin
append-path PATH /grid/scp/hw/tools/arm-none-eabi/8.3.0/armv7m-none-eabi/bin
append-path PATH /grid/scp/hw/tools/arm-none-eabi/8.3.0/aarch64-none-elf/bin
```

2.3.8. arm-none-eabi-cov/4.8

```
append-path PATH /grid/scp/hw/tools/arm-none-eabi-cov/4.8/bin
```

2.3.9. arm-linux-gnu/8.3-2019.03

```
append-path PATH /grid/scp/hw/tools/arm-linux-gnu/8.3-2019.03/gcc-arm-8.3-2019.03-x86_64-arm-linux-gnueabi/bin
append-path PATH /grid/scp/hw/tools/arm-linux-gnu/8.3-2019.03/gcc-arm-8.3-2019.03-x86_64-aarch64-linux-gnu/bin
```

2.3.10. socrates/sysoc/1.2.0

```
set  cds_root           /grid/scp/hw/tools
set   cds_pkg            Socrates/SYSOC-1.2.0
set             doulog_root          $cds_root/$cds_pkg
setenv          CLC_VERSION             $doulog_root
append-path  PATH    $doulog_root
```

2.3.11. stratus/162/16.21.100

```
set             cds_root          /grid/scp/hw/tools
set             cds_pkg           STRATUS162/16.21.100
append-path     PATH              $cds_root/$cds_pkg/bin
```

2.3.12. blueprint/3.7.5/3.7.5

```
set             cds_root           /grid/scp/hw/tools
set             cds_pkg            blueprint/3.7.5
setenv          BLUEPRINT_HOME  $cds_root/$cds_pkg
setenv          BP_GEN_HOME   $cds_root/$cds_pkg/blueprint_generators
append-path     PATH               $cds_root/$cds_pkg/bin
```

2.3.13. parasoft/cpptest/2022.1

```
set ThisVersion "v2022.1"
set AppName "Parasoft CPPtest"
set InstallPath "/grid/common/pkgs/cpptest"
set   ApplicationPath           "$InstallPath/$ThisVersion"
prepend-path    PATH            "$ApplicationPath"
```

2.3.14. fop/1.0

```
prepend-path PATH /grid/scp/hw/tools/fop/fop-1.0
```

2.3.15. python/2.7.2

```
set ThisVersion "v2.7.2"
set AppName "python"
set InstallPath "/grid/common/pkgs"
set   ApplicationPath           "$InstallPath/$AppName/$ThisVersion"
prepend-path    PATH            "$ApplicationPath/bin"
prepend-path    MANPATH         "$ApplicationPath/man"
prepend-path    LD_LIBRARY_PATH "$ApplicationPath/lib"
```

2.3.16. ldra/toolsuite/9.7.1

```
set             ldra_version            9.7.1
set             cds_root                /grid/scp/hw/tools
set             cds_pkg                 LDRA/$ldra_version/Toolsuite
setenv          LDRA_LICENSE_FILE   xxxx
append-path     PATH                    $cds_root/$cds_pkg
append-path     LM_LICENSE_FILE   xxxx
```

2.3.17. ti-cgt-arm/16.9.4.LTS

```
set             cds_root                /grid/scp/hw/tools
set             cds_pkg                 ti-cgt-arm/16.9.4.LTS
setenv          TI_CGT_ARM_INCLUDE_DIR  $cds_root/$cds_pkg/include
append-path     PATH                    $cds_root/$cds_pkg/bin
```

2.3.18. gcc/6.3.0

```
prepend-path    PATH    /grid/common/pkgs/gcc/v6.3.0/bin
```

2.3.19. clang/6.0.0

```
prepend-path    PATH    /grid/scp/hw/tools/clang/6.0.0/bin
```

2.3.20. uncrustify/v0.60

```
set ThisVersion "v0.60"
set AppName "uncrustify"
set InstallPath "/grid/common/pkgs"
set   ApplicationPath           "$InstallPath/$AppName/$ThisVersion"
prepend-path    PATH            "$ApplicationPath/bin"
prepend-path    MANPATH         "$ApplicationPath/man"
prepend-path    LD_LIBRARY_PATH "$ApplicationPath/lib"
```

2.3.21. doxygen/1.8.9.1

```
set ThisVersion "v1.8.9.1"
set AppName "doxygen"
set InstallPath "/grid/common/pkgs"
set   ApplicationPath           "$InstallPath/$AppName/$ThisVersion"
prepend-path    PATH            "$ApplicationPath/bin"
prepend-path    MANPATH         "$ApplicationPath/man"
prepend-path    LD_LIBRARY_PATH "$ApplicationPath/lib"
```

2.3.22. perl/5.8.8

```
set ThisVersion "v5.8.8"
set AppName "perl"
set InstallPath "/grid/common/pkgs"
set   ApplicationPath           "$InstallPath/$AppName/$ThisVersion"
prepend-path    PATH            "$ApplicationPath/bin"
prepend-path    MANPATH         "$ApplicationPath/man"
prepend-path    LD_LIBRARY_PATH "$ApplicationPath/lib"
```

2.3.23. sia/171/17.10.001-p-382

```
set             cds_root         /grid/avs/pkgs/glue/SIA
set             cds_pkg          sia_17.10.001-p-382
set             app              $cds_root/$cds_pkg
setenv TTI_IPXACT_TOOL_PATH $app/sia/bin
setenv          SIA_HOME        $app
append-path     PATH            $app/sia/bin
append-path     LM_LICENSE_FILE   xxxx
```

Chapter 3. Test Scenarios

3.1. Overview

In order to verify, that driver works as expected with the controller the following tests were used. All tests were executed on RTL and/or TLM environment (where applicable). All results should be the same on both platforms (it is allowed to modify/configure the test case between RTL and TLM, (to match the test environment), however it is not allowed to modify/configure the driver code). Once test criteria are met, the test is marked as passed. Otherwise test is marked as failed. In some cases, test is skipped. This is because test requires some controller's feature which is not supported by current configuration of controller IP or the test platform. Note, that some test scenarios are positive (some kind of action must be done) and some of them are negative (some kind of error must occur).

3.2. List of test scenarios

3.2.1. Functional tests

3.2.1.1. test_sdr_master_sec_master_tx

Test Description:	This test transfers data from master to secondary master
Pass Conditions:	data transfer is success
Fail Conditions:	data mismatch
APIs called:	I3C_SlaveModeReqSdrRead, I3C_CmdClearAll, I3C_CmdAddPrivWrite, I3C_CmdExec
Covered Use Cases:	I3C bus management, Interrupts, SDR mode support

3.2.1.2. test_sdr_master_sec_master_rx

Test Description:	This test transfers data from secondary master to master
Pass Conditions:	data transfer is success
Fail Conditions:	data mismatch
APIs called:	I3C_SlaveModeReqSdrRead, I3C_CmdClearAll, I3C_CmdAddPrivWrite, I3C_CmdExec
Covered Use Cases:	I3C bus management, Interrupts, SDR mode support

3.2.1.3. test_sdr_master_sec_master_tx_multiple

Test Description:	This test transfers data from master to secondary master multiple times
Pass Conditions:	data transfer is success
Fail Conditions:	data mismatch
APIs called:	I3C_SlaveModeReqSdrRead, I3C_CmdClearAll, I3C_CmdAddPrivWrite, I3C_CmdExec
Covered Use Cases:	I3C bus management, Interrupts, SDR mode support

3.2.1.4. test_sdr_master_sec_master_tx_threshold

Test Description:	This test transfers data to secondary master using threshold mechanism
Pass Conditions:	data transfer is success
Fail Conditions:	data mismatch
APIs called:	I3C_SlaveModeReqSdrRead, I3C_CmdClearAll, I3C_CmdAddPrivWrite, I3C_CmdExec
Covered Use Cases:	I3C bus management, Interrupts, SDR mode support

3.2.1.5. test_sdr_master_sec_master_rx_threshold

Test Description:	This test transfers data from secondary master using threshold mechanism
Pass Conditions:	data transfer is success
Fail Conditions:	data mismatch
APIs called:	I3C_SlaveModeReqSdrRead, I3C_CmdClearAll, I3C_CmdAddPrivWrite, I3C_CmdExec
Covered Use Cases:	I3C bus management, Interrupts, SDR mode support

3.2.1.6. test_sdr_master_slave_tx

Test Description:	This test transfers data to slave
Pass Conditions:	data transfer is success
Fail Conditions:	data mismatch
APIs called:	I3C_SlaveModeReqSdrRead, I3C_CmdClearAll, I3C_CmdAddPrivWrite, I3C_CmdExec
Covered Use Cases:	I3C bus management, Interrupts, SDR mode support

3.2.1.7. test_sdr_master_slave_rx

Test Description:	This test transfers data from slave
Pass Conditions:	data transfer is success
Fail Conditions:	data mismatch
APIs called:	I3C_SlaveModeReqSdrRead, I3C_CmdClearAll, I3C_CmdAddPrivWrite, I3C_CmdExec
Covered Use Cases:	I3C bus management, Interrupts, SDR mode support

3.2.1.8. test_mastership_request_sec_mster

Test Description:	This test tries to request mastership as a secondary master
Pass Conditions:	secondary master changed operation mode

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdAddDefineSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus, I3C_CmdAddGetAccMst, I3C_ConfigureSlaveInterrupts, I3C_SlaveModeConfigure, I3C_SlaveModeReqSdrRead, I3C_SlaveModeReqSdrWrite, I3C_SlaveModeReqDdrRead, I3C_SlaveModeReqDdrWrite, I3C_SlaveModeRequestHotJoin, I3C_SlaveModeMastershipReq, I3C_GetAsfInfo, I3C_CheckOperationMode

Covered Use Cases: I3C Bus management, Mastership request, Slave mode functions

3.2.1.9. test_cmd_list_init

Test Description: This test checks if command list can be initialized and destroyed. Checks also if error is returned when cmd_list is NULL

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: internal functions only

Covered Use Cases: none

3.2.1.10. test_cmd_list_max_fill

Test Description: This test tries to fill list to maximum

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: internal functions only

Covered Use Cases: none

3.2.1.11. test_cmd_list_fill_and_empty_pop

Test Description: This tests tries ties to fill list to maximum, gets back all data by id, deletes it and checks data integrity

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: internal functions only

Covered Use Cases: none

3.2.1.12. test_cmd_list_try_to_overfill

Test Description: This test checks if list can be overfilled, also checks if will report an error in case of write to full list

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: internal functions only

Covered Use Cases: none

3.2.1.13. test_cmd_list_data_integrity_of_overfilled

Test Description: This test checks if buffer is FIFO. Test will try to overfill the buffer. Checks if data firs written is data firs read

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: internal functions only

Covered Use Cases: none

3.2.1.14. test_cmd_list_try_to_overread

Test Description: This test tries to delete too much data from command list. Checks if list reports an error in case of no elements available

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: internal functions only

Covered Use Cases: none

3.2.1.15. test_cmd_list_search

Test Description: This test tries to find element of the list by command ID. Checks also if error is returned when list is empty

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: internal functions only

Covered Use Cases: none

3.2.1.16. test_cmd_list_delete_any

Test Description: This test tries to delete element from random place of the list. Checks also if error is returned when list is empty

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: internal functions only

Covered Use Cases: none

3.2.1.17. test_cmd_list_traverse

Test Description: This test tries to traverse list and call callback function on every available element

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: internal functions only

Covered Use Cases: none

3.2.1.18. test_init_params

Test Description: This test checks if initial parameters are correct

Pass Conditions: parameters are correct

Fail Conditions: any other status

APIs called: I3C_Probe, I3C_Init, I3C_Start, I3C_EnableCore, I3C_DisableCore

Covered Use Cases: Common architecture, Driver auto configuration

3.2.1.19. test_init_sec_master_disable_interrupts

Test Description: This test tries to initialize secondary master with all interrupts disabled

Pass Conditions: interrupts are configured properly

Fail Conditions: any other status

APIs called: I3C_Probe, I3C_Init, I3C_Start, I3C_EnableCore, I3C_DisableCore

Covered Use Cases: Common architecture, Driver auto configuration

3.2.1.20. test_init_parts_core_enabled

Test Description: This test tries to execute functions which requires core to be disabled

Pass Conditions: core is enabled

Fail Conditions: any other status

APIs called: I3C_Probe, I3C_Init, I3C_Start, I3C_EnableCore, I3C_DisableCore

Covered Use Cases: Common architecture, Driver auto configuration

3.2.1.21. test_init_stop_interrupt

Test Description: This test tries to call API isr implementation when core has been stopped

Pass Conditions: interrupt is not handled

Fail Conditions: any other status

APIs called: I3C_Probe, I3C_Isr, I3C_Init, I3C_Start, I3C_EnableCore, I3C_DisableCore

Covered Use Cases: Common architecture, Driver auto configuration

3.2.1.22. test_init_destroy

Test Description: This test tries to destroy driver

Pass Conditions: driver destroyed

Fail Conditions: any other status

APIs called: I3C_Probe, I3C_Init, I3C_Start, I3C_EnableCore, I3C_DisableCore

Covered Use Cases: Common architecture, Driver auto configuration

3.2.1.23. test_init_bus_mode

Test Description: This test checks if correct bus mode is set

Pass Conditions: correct bus mode

Fail Conditions: any other status

APIs called: I3C_Probe, I3C_Init, I3C_Start, I3C_EnableCore, I3C_DisableCore

Covered Use Cases: Common architecture, Driver auto configuration

3.2.1.24. test_init_without_devs

Test Description: This test tries to initialize master without devices on the bus

Pass Conditions: master not initialized

Fail Conditions: any other status

APIs called: I3C_Probe, I3C_Init, I3C_Start, I3C_EnableCore, I3C_DisableCore

Covered Use Cases: Common architecture, Driver auto configuration

3.2.1.25. test_init_core_idle

Test Description: This test injects error, core cannot be disabled

Pass Conditions: core not disabled, CDN_EIO returned

Fail Conditions: any other status

APIs called: I3C_Probe, I3C_Init, I3C_Start, I3C_EnableCore, I3C_DisableCore, I3C_ConfigureInterrupts, I3C_ConfigureThresholds

Covered Use Cases: Common architecture, Interrupts

3.2.1.26. test_init_too_many_devs

Test Description: This test tries to initialize master with too many devices

Pass Conditions: master not initialized

Fail Conditions: any other status

APIs called: I3C_Probe, I3C_Init, I3C_Start, I3C_EnableCore, I3C_DisableCore, I3C_ConfigureInterrupts, I3C_ConfigureThresholds

Covered Use Cases: Common architecture, Interrupts

3.2.1.27. test_init_pure_bus

Test Description: This test initializes driver with HDR capable devices only

Pass Conditions: bus is pure after initialization

Fail Conditions: any other status

APIs called: I3C_Probe, I3C_Init, I3C_Start, I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetStatus

Covered Use Cases: Common architecture, Various I3C Bus modes, I3C Bus management

3.2.1.28. test_init_prescalers_with_no_timings

Test Description: This test tries to reconfigure prescalers

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: I3C_Probe, I3C_Init, I3C_Start, I3C_SetBusMode, I3C_GetBusMode, I3C_ConfigurePrescalers, I3C_DevPrint

Covered Use Cases: Common architecture, Various I3C Bus modes, Prescaler configuration

3.2.1.29. test_ibi

Test Description: This test performs In-Band Interrupt

Pass Conditions: ibi is acked

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus, I3C_IbiConfigureDevices, I3C_IbiModifyDeviceConfig, I3C_IbiGetAddressOfIssuer, I3C_IbiGetData

Covered Use Cases: I3C Bus management, Interrupts, In-Band Interrupts

3.2.1.30. test_ibi_nack

Test Description: This test performs In-Band Interrupt when IBI is disabled

Pass Conditions: ibi is nacked

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus, I3C_IbiConfigureDevices, I3C_IbiModifyDeviceConfig, I3C_IbiGetAddressOfIssuer, I3C_IbiGetData

Covered Use Cases: In-Band Interrupts

3.2.1.31. test_ibi_address_of_issuer

Test Description: This test injects error into SIR MAP registers

Pass Conditions: correct addresses are returned

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus, I3C_IbiConfigureDevices, I3C_IbiModifyDeviceConfig, I3C_IbiGetAddressOfIssuer, I3C_IbiGetData

Covered Use Cases: In-Band Interrupts

3.2.1.32. test_ibi_slot_overflow

Test Description: This test tries to overflow IBI slots

Pass Conditions: API function returns CDN_EINVAL

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus, I3C_IbiConfigureDevices, I3C_IbiModifyDeviceConfig, I3C_IbiGetAddressOfIssuer, I3C_IbiGetData

Covered Use Cases: In-Band Interrupts

3.2.1.33. test_ibi_tcam0_event

Test Description: This test tries to overflow IBI slots

Pass Conditions: API function returns CDN_EINVAL

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr,

I3C_CmdAddGetDcr, I3C_CmdAddGetStatus, I3C_IbiConfigureDevices, I3C_IbiModifyDeviceConfig, I3C_IbiGetAddressOfIssuer, I3C_IbiGetData

Covered Use Cases: In-Band Interrupts

3.2.1.34. test_getasfinfo

Test Description: This test checks whether ASF info is provided

Pass Conditions: asf info provided

Fail Conditions: any other status

APIs called: I3C_GetAsfInfo

Covered Use Cases: ASF fault events

3.2.1.35. test_cmdaddddrread

Test Description: This test checks ID range of DDR IDs in for read operation

Pass Conditions: id checked as expected

Fail Conditions: any other status

APIs called: I3C_CmdAddDdrRead

Covered Use Cases: Transfer data using DDR mode

3.2.1.36. test_cmdaddddrwrite

Test Description: This test checks ID range of DDR IDs in for write operation

Pass Conditions: id checked as expected

Fail Conditions: any other status

APIs called: I3C_CmdAddDdrWrite

Covered Use Cases: Transfer data using DDR mode

3.2.1.37. test_cmdexec

Test Description: This test checks whether CMD_IN_PROGRESS flag is checked properly

Pass Conditions: status is checked as expected

Fail Conditions: any other status

APIs called: I3C_CmdExec

Covered Use Cases: Commands support

3.2.1.38. test_non_imm_bus_init

Test Description: This test initializes the bus using non-immediate commands

Pass Conditions: data transfer after initialization passed

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C Bus management, Immediate commands, Prescaler configuration

3.2.1.39. test_imm_bus_init

Test Description: This test initializes the bus using immediate commands

Pass Conditions: data transfer after initialization passed

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C Bus management, Immediate commands, Prescaler configuration

3.2.1.40. test_imm_non_imm_mix_bus_init

Test Description: This test initializes the bus using mixed immediate and non-immediate commands

Pass Conditions: data transfer after initialization passed

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C Bus management, Immediate commands, Prescaler configuration

3.2.1.41. test_imm_in_progress

Test Description: This test tries to send immediate command when other immediate command is in progress

Pass Conditions: API function returns CDN_EBUSY

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: Immediate commands

3.2.1.42. test_imm_inject

Test Description: This test injects immediate command between non-immediate commands

Pass Conditions: immediate command is execute first

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C Bus management, Immediate commands

3.2.1.43. test_i2c_read

Test Description: This test transfer data from I2C device

Pass Conditions: data transfer is success

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_CmdAddPrivI2CWrite, I3C_CmdAddPrivI2CRead, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C bus management, I2C legacy devices

3.2.1.44. test_i2c_write

Test Description: This test transfer data to I2C device

Pass Conditions: data transfer is success

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_CmdAddPrivI2CWrite, I3C_CmdAddPrivI2CRead, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C bus management, I2C legacy devices

3.2.1.45. test_i2c_inactive_device_init

Test Description: This test tries to initialize inactive I2C device

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: I3C_Probe, I3C_Init, I3C_Start, I3C_SetBusMode, I3C_EnableCore, I3C_GetBusMode, I3C_DevPrint

Covered Use Cases: I2C legacy devices

3.2.1.46. test_ccc_set_new_da

Test Description: This test updates Dynamic Address of the device

Pass Conditions: device has correct address

Fail Conditions: device has not correct address

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddSetNewDa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C Bus management

3.2.1.47. test_ccc_set_get_max_read_length

Test Description: This test set maximum data read length

Pass Conditions: read correct value from the device

Fail Conditions: value read from device is incorrect

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddSetNewDa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C Bus management, Interrupts

3.2.1.48. test_ccc_set_get_max_read_length_bcst_with_ibi

Test Description: This test set maximum data read length to all devices

Pass Conditions: read correct value from the device

Fail Conditions: value read from device is incorrect

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddSetNewDa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C Bus management, Interrupts

3.2.1.49. test_ccc_set_get_max_write_length

Test Description: This test set maximum data write length

Pass Conditions: read correct value from the device

Fail Conditions: value read from device is incorrect

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddSetNewDa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C Bus management, Interrupts

3.2.1.50. test_ccc_set_get_max_write_length_bcst

Test Description: This test set maximum data write length to all devices

Pass Conditions: read correct value from the device

Fail Conditions: value read from device is incorrect

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddSetNewDa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C Bus management, Interrupts

3.2.1.51. test_cmd_buffer_overflow

Test Description: This test tries to add CCC commands when software FIFO is full

Pass Conditions: API function returns CDN_EBUSY

Fail Conditions: API function does not return CDN_EBUSY

APIs called: I3C_CmdAddDdrRead, I3C_CmdAddDdrWrite, I3C_CmdAddSetSlaveEvents, I3C_CmdAddEnterActivityState, I3C_CmdAddResetDaa, I3C_CmdAddEnterDaa, I3C_CmdAddSetMaxWriteLength, I3C_CmdAddGetMaxWriteLength, I3C_CmdAddSetMaxReadLength, I3C_CmdAddGetMaxReadLength, I3C_CmdAddDefineSlavesList, I3C_CmdAddEnterTestMode, I3C_CmdAddEnterHdrMode, I3C_CmdSetDaFromSa, I3C_CmdAddSetNewDa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus, I3C_CmdAddGetAccMst, I3C_CmdAddPrivRead, I3C_CmdAddPrivWrite, I3C_CmdAddGetMaxDataSpeed

Covered Use Cases: Software command queue

3.2.1.52. test_ccc_set_ad_from_sa_no_device

Test Description: This test updates Dynamic Address of non-existing device

Pass Conditions: API function returns CDN_EINVAL

Fail Conditions: API function does not return CDN_EINVAL

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList,

I3C_CmdSetDaFromSa, I3C_CmdAddSetNewDa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: Change address of dynamic addressed device

3.2.1.53. test_cmd_large_without_fifos

Test Description: This test tries to send command with large payload without data FIFOs

Pass Conditions: API function returns CDN_EINVAL

Fail Conditions: API function does not return CDN_EINVAL

APIs called: I3C_CmdExec, I3C_CmdExecImmediate, I3C_EnableMcs, I3C_DisableMcs, I3C_ManualCommandStart

Covered Use Cases: Commands support

3.2.1.54. test_ccc_get_status

Test Description: This test get status of the device

Pass Conditions: Status is correct

Fail Conditions: Any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_CmdExec, I3C_CmdExecImmediate, I3C_CmdExecImmediate, I3C_EnableMcs, I3C_DisableMcs, I3C_ManualCommandStart, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C bus management, Commands Interrupts, Commands support

3.2.1.55. test_ccc_enter_test_mode

Test Description: This test sends enter test mode to slave

Pass Conditions: slave got test mode event

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_CmdExec, I3C_CmdExecImmediate, I3C_EnableMcs, I3C_DisableMcs, I3C_ManualCommandStart, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdAddEnterTestMode, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C bus management, Commands Interrupts, Commands support, Test Mode

3.2.1.56. test_ccc_reset_daa_exec_wait

Test Description: This test resets DA of the device

Pass Conditions: device cannot perform transfer

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddResetDaa, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C bus management, Commands Interrupts

3.2.1.57. test_ccc_reset_daa_broadcast

Test Description: This test resets DA of all devices

Pass Conditions: device cannot perform transfer

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddResetDaa, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C bus management, Commands Interrupts

3.2.1.58. test_ccc_enter_activity_state

Test Description: This test sends enter activity state mode to slave

Pass Conditions: correct status is returned from slave

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterActivityState, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C bus management, Commands Interrupts, Power Management

3.2.1.59. test_ccc_set_slave_events

Test Description: This test disabled IBI event

Pass Conditions: slave cannot perform IBI

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C bus management, Commands Interrupts, IBI Interrupts

3.2.1.60. test_ccc_set_get_max_data_speed

Test Description: This test set maximum data speed

Pass Conditions: correct speed is returned from slave

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddResetDaa, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C bus management, Commands Interrupts

3.2.1.61. test_ccc_slave_event_invalid_mask

Test Description: This test tries to set invalid slave event mask

Pass Conditions: API function returns CDN_EPROTO

Fail Conditions: any other status

APIs called: I3C_CmdExec, I3C_CmdExecImmediate, I3C_EnableMcs, I3C_DisableMcs, I3C_ManualCommandStart

Covered Use Cases: Commands Support

3.2.1.62. test_ccc_set_nca_mode_in_priv_write

Test Description: This test set NCA mode for private write operation in slave mode

Pass Conditions: slave cannot perform IBI

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C vendor specific extension for data pattern with or without sub framing

3.2.1.63. test_ccc_set_slave_mrl_mwl

Test Description: This test set MRL and MWL values for a slave

Pass Conditions: slave changes MWL and MRL values

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C vendor specific extension for data pattern with or without sub framing

3.2.1.64. test_ccc_set_slave_buscon

Test Description: This test set storage data and fill level for SETBUSCON CCC

Pass Conditions: slave changes BUSCON values in status registers

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C vendor specific extension for data pattern with or without sub framing

3.2.1.65. test_ccc_set_target_reset_broadcast

Test Description: This test send resest command to all targets

Pass Conditions: slave target send ack to master

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C vendor specific extension for data pattern with or without sub framing

3.2.1.66. test_ccc_set_time_control

Test Description: This test enable Async Mode0 for a slave

Pass Conditions: Slave enable async mode0

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C vendor specific extension for data pattern with or without sub framing

3.2.1.67. test_ccc_get_time_control

Test Description: This test reads TCAM0 data from a slave

Pass Conditions: Read data from slave successfully

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C vendor specific extension for data pattern with or without sub framing

3.2.1.68. test_ccc_set_group_address

Test Description: This test set group address for slave or secondary master

Pass Conditions: Set group address correctly

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C vendor specific extension for data pattern with or without sub framing

3.2.1.69. test_ccc_set_define_group_list

Test Description: This test set group address for multiple targets

Pass Conditions: Group address sets correctly for multiple targets

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C vendor specific extension for data pattern with or without sub framing

3.2.1.70. test_ccc_reset_group_address

Test Description: This test remove group address

Pass Conditions: group address removed for the target

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C vendor specific extension for data pattern with or without sub framing

3.2.1.71. test_isr_no_interrupt

Test Description: This test tries to call API isr implementation when interrupt did not occured

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: I3C_Probe, I3C_Init, I3C_Start, I3C_Isr

Covered Use Cases: Interrupts

3.2.1.72. test_isr_tx_rx_thr_cmd_list_empty

Test Description: This test tries to call API isr implementation with threshold interrupts

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: I3C_Probe, I3C_Init, I3C_Start, I3C_Isr

Covered Use Cases: Interrupts

3.2.1.73. test_auto_configuration

Test Description: This test tries to inject error in auto-configuration feature

Pass Conditions: configuration is reflected in private data

Fail Conditions: any other status

APIs called: I3C_Probe, I3C_Init, I3C_Start

Covered Use Cases: Driver auto configuration

3.2.1.74. test_transfer_to_non_existing_device

Test Description: This test performs transfer to non-existing device

Pass Conditions: API function returns CDN_EINVAL

Fail Conditions: any other status

APIs called: I3C_Probe, I3C_Init, I3C_Start, I3C_CmdAddPrivRead, I3C_CmdAddPrivWrite, I3C_CmdAddDdrWrite, I3C_CmdAddDdrRead

Covered Use Cases: Change address of dynamic addressed device

3.2.1.75. test_hot_join_sec_mster

Test Description: This test adds new secondary master device to the bus

Pass Conditions: appropriate device joined the bus

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa,

I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus, I3C_HjConfigureResponse

Covered Use Cases: I3C Bus management, Hotjoin

3.2.1.76. test_hot_join_slave

Test Description: This test adds new slave device to the bus

Pass Conditions: appropriate device joined the bus

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus, I3C_HjConfigureResponse

Covered Use Cases: I3C Bus management, Hotjoin

3.2.1.77. test_bytes_swap_3_bytes

Test Description: Test Bytes swap - 3 bytes

Pass Conditions: Array of bytes is reversed as expected.

Fail Conditions: All other status

APIs called: internal functions only

Covered Use Cases: none

3.2.1.78. test_bytes_swap_4_bytes

Test Description: Test Bytes swap - 4 bytes

Pass Conditions: Array of bytes is reversed as expected.

Fail Conditions: All other status

APIs called: internal functions only

Covered Use Cases: none

3.2.1.79. test_sanity_callbacks

Test Description: This test checks whether sanity functions check input value

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: internal functions only

Covered Use Cases: none

3.2.1.80. test_sanity_function2

Test Description: This test checks whether internal function I3C_SanityFunction2 checks input values

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: internal functions only

Covered Use Cases: none

3.2.1.81. test_sanity_function9

Test Description: This test checks whether internal function I3C_SanityFunction9 checks input values

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: internal functions only

Covered Use Cases: none

3.2.1.82. test_sanity_function30

Test Description: This test checks whether internal function I3C_SanityFunction30 checks input values

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: internal functions only

Covered Use Cases: none

3.2.1.83. test_sanity_function35

Test Description: This test checks whether internal function I3C_SanityFunction35 checks input values

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: internal functions only

Covered Use Cases: none

3.2.1.84. test_sanity_function37

Test Description: This test checks whether internal function I3C_SanityFunction38 checks input values

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: internal functions only

Covered Use Cases: none

3.2.1.85. test_sanity_function48

Test Description: This test checks whether internal function I3C_SanityFunction50 checks input values

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: internal functions only

Covered Use Cases: none

3.2.1.86. test_sanity_function60

Test Description: This test checks whether internal function I3C_SanityFunction70 checks input values

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: internal functions only

Covered Use Cases: none

3.2.1.87. test_sanity_function69

Test Description: This test checks whether internal function I3C_SanityFunction93 checks input values

Pass Conditions: operation passed

Fail Conditions: any other status

APIs called: internal functions only

Covered Use Cases: none

3.2.1.88. test_ddr_master_sec_master_tx

Test Description: This test transfers data from master to secondary master in HDR mode

Pass Conditions: data transfer is success

Fail Conditions: data mismatch

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddDdrWrite, I3C_CmdAddDdrRead, I3C_CmdAddEnterActivityState, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdAddEnterHdrMode, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C Bus management, Interrupts, DDR mode

3.2.1.89. test_ddr_master_sec_master_tx_threshold

Test Description: This test transfers data from master to secondary master in HDR mode

Pass Conditions: data transfer is success

Fail Conditions: data mismatch

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddDdrWrite, I3C_CmdAddDdrRead, I3C_CmdAddEnterActivityState, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdAddEnterHdrMode, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C Bus management, Interrupts, DDR mode

3.2.1.90. test_ddr_master_sec_master_rx

Test Description: This test transfers data from secondary master to master in HDR mode

Pass Conditions: data transfer is success

Fail Conditions: data mismatch

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddDdrWrite, I3C_CmdAddDdrRead, I3C_CmdAddEnterActivityState, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdAddEnterHdrMode, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C Bus management, Interrupts, DDR mode

3.2.1.91. test_ddr_master_sec_master_rx_threshold

Test Description: This test transfers data from secondary master to master using threshold mechanism in HDR mode

Pass Conditions: data transfer is success

Fail Conditions: data mismatch

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddDdrWrite, I3C_CmdAddDdrRead, I3C_CmdAddEnterActivityState, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdAddEnterHdrMode, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C Bus management, Interrupts, DDR mode

3.2.1.92. test_ddr_master_slave_tx

Test Description: This test transfers data to slave in HDR mode

Pass Conditions: data transfer is success

Fail Conditions: data mismatch

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddDdrWrite, I3C_CmdAddDdrRead, I3C_CmdAddEnterActivityState, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdAddEnterHdrMode, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C Bus management, Interrupts, DDR mode

3.2.1.93. test_ddr_master_slave_rx

Test Description: This test transfers data from slave in HDR mode

Pass Conditions: data transfer is success

Fail Conditions: data mismatch

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddDdrWrite, I3C_CmdAddDdrRead, I3C_CmdAddEnterActivityState, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdAddEnterHdrMode, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C Bus management, Interrupts, DDR mode

3.2.1.94. test_sec_master_tx_no_data

Test Description: This test tries to transfer data when payload is empty

Pass Conditions: data transfer is success

Fail Conditions: data mismatch

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddDdrWrite, I3C_CmdAddDdrRead, I3C_CmdAddEnterActivityState, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdAddEnterHdrMode, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: I3C Bus management, Interrupts, DDR mode

3.2.1.95. test_sec_master_rx_without_daa

Test Description: This test tries to transfer data without Dynamic Address

Pass Conditions: API function returns CDN_ENOTSUP/otherwise

Fail Conditions: any other status

APIs called: I3C_SetBcr, I3C_SetDcr, I3C_SetPid, I3C_ConfigureDevices, I3C_ConfigureDevice, I3C_ClearRrOfDevice, I3C_CmdAddDdrWrite, I3C_CmdAddDdrRead, I3C_CmdAddEnterActivityState, I3C_CmdAddEnterDaa, I3C_GetSlavesList, I3C_CmdAddEnterHdrMode, I3C_CmdSetDaFromSa, I3C_CmdAddGetProvisionalId, I3C_CmdAddGetBcr, I3C_CmdAddGetDcr, I3C_CmdAddGetStatus

Covered Use Cases: DDR mode

3.2.1.96. test_get_slave_mrl_mwl

Test Description: This test reads mrl and mwl value from slave

Pass Conditions: mrl and mwl values read correctly

Fail Conditions: any other status Use Case

Covered Use Cases: I3C Bus management, status read from target

3.2.1.97. test_sdr_master_slave_fill_lvl

Test Description: This test reads fill level from slave

Pass Conditions: mrl and mwl values read correctly

Fail Conditions: any other status Use Case

Covered Use Cases: I3C Bus management, status read from target

3.2.1.98. test_sdr_slave_flush_done

Test Description: This test initiate flush at Target side

Pass Conditions: flush data from fifo

Fail Conditions: any other status Use Case

Covered Use Cases: I3C Bus management, status read from target

3.2.1.99. Test BIST1

Test Description: Test uses ASF_SelfTest function and is intended to checking whether all emulated event will be reported.

Pass Conditions: ASF is supported.

Fail Conditions: All other status

APIs called: ASF_SelfTest

Covered Use Cases: ASF self test

3.2.1.100. Test BIST2

Test Description: Test uses function ASF_TestEvent to trigger all emulated fault events.

Pass Conditions: ll events should be generated as expected.

Fail Conditions: All other status

APIs called: ASF_EnableAllEvents, ASF_TestEvent, ASF_CheckIfASFSupported, ASF_GetStatistic, ASF_DisableAllEvents

Covered Use Cases: Information about fault events, ASF Interrupts

3.2.1.101. ASF_DisableAllEvents

Test Description: Tests if DisableAllEvents function work correct. Test disable detection of all events, then starts generating of all event by using ASF_TestEvent function. For each emulated event function should receive timeout error.

Pass Conditions: No event should be generated.

Fail Conditions: All other status

APIs called: ASF_DisableAllEvents, ASF_CheckIfASFSupported, ASF_TestEvent

Covered Use Cases: ASF Interrupts, ASF Self Test

3.2.1.102. ASF_EnableAllEvents

Test Description: Function disables detection all events then checks if all events were disabled. In next step enables all events. After enabling events, function trigger all events and checks whether they were reported

Pass Conditions: All events should be generated.

Fail Conditions: All other status

APIs called: ASF_EnableAllEvents, ASF_CheckIfASFSupported, ASF_TestEvent

Covered Use Cases: ASF Interrupts, ASF Self Test

3.2.1.103. ASF_EnableEvent

Test Description: Tests ASF_EnableEvent function. At the beginning function disables all events. In next step in loop enables single event, trigger enabled event and checks if correct event was detected. In nested loop function tests if all disabled events returns CDN_EPROTO error code.

Pass Conditions: A single event should be generated.

Fail Conditions: All other status

APIs called: ASF_EnableEvent, ASF_DisableEvent, ASF_CheckIfASFSupported, ASF_TestEvent

Covered Use Cases: Information about fault event, ASF Interrupts

3.2.1.104. ASF_SetEventAsNonFatal

Test Description: Function tests ASF_SetEventAsNonFatal function for given instance. Function enables and sets all events as fatal. Then one by one set single event as non fatal and trigger events, and checks if generated events are correct.

Pass Conditions: A single event should be generated as non fatal.

Fail Conditions: All other status

APIs called: ASF_SetEventAsFatal, ASF_SetEventAsNonFatal, ASF_CheckIfASFSupported, ASF_TestEvent

Covered Use Cases: Information about fault event, ASF Interrupts

3.2.1.105. ASF_DisableEvent

Test Description: Tests ASF_DisableEvent function. At the beginning function enable all events. Then disables one by one only single event and verify if it was disabled.

Pass Conditions: A single event should be disabled.

Fail Conditions: All other status

APIs called: ASF_DisableEvent, ASF_EnableEvent, ASF_CheckIfASFSupported, ASF_TestEvent

Covered Use Cases: Information about fault event, ASF Interrupts

3.2.1.106. ASF_EnableProtocolEventByMask

Test Description: Tests ASF_EnableProtocolEventByMask function.

Pass Conditions: All events should be enabled.

Fail Conditions: All other status

APIs called: ASF_GetSupportedProtocolErrors, ASF_SetEventAsFatal, ASF_DisableProtocolEventByMask, ASF_EnableProtocolEventByMask

Covered Use Cases: Information about fault event, ASF Configuration

3.2.1.107. ASF_DisableProtocolEventByMask

Test Description: Tests ASF_DisableProtocolEventByMask function.

Pass Conditions: No events should be enabled.

Fail Conditions: All other status

APIs called: ASF_GetSupportedProtocolErrors, ASF_SetEventAsFatal, ASF_DisableProtocolEventByMask, ASF_EnableProtocolEventByMask

Covered Use Cases: Information about fault event, ASF Configuration

3.2.1.108. ASF_EnableProtocolEventByIDFunc

Test Description: Tests ASF_EnableProtocolEventByIDFunc function.

Pass Conditions: All events should be enabled.

Fail Conditions: All other status

APIs called: ASF_GetSupportedProtocolErrors, ASF_SetEventAsFatal, ASF_DisableProtocolEventByMask, ASF_EnableProtocolEventByID

Covered Use Cases: Information about fault event, ASF Configuration

3.2.1.109. ASF_DisableProtocolEventByIDFunc

Test Description: Tests ASF_DisableProtocolEventByIDFunc function.

Pass Conditions: All events should be disabled.

Fail Conditions: All other status

APIs called: ASF_GetSupportedProtocolErrors, ASF_SetEventAsFatal, ASF_EnableProtocolEventByMask, ASF_DisableProtocolEventByID

Covered Use Cases: Information about fault event, ASF Configuration

3.2.1.110. ASF_EnableTimoutEventByMask

Test Description: Tests ASF_EnableTimeoutEventByMask function.

Pass Conditions: All events should be enabled.

Fail Conditions: All other status

APIs called: ASF_GetSupportedTimeoutErrors, ASF_SetEventAsFatal, ASF_EnableTimeoutEventByMask, ASF_DisableTimeoutEventByMask

Covered Use Cases: Information about fault event, ASF Configuration

3.2.1.111. ASF_DisableTimoutEventByMask

Test Description: Tests ASF_DisableTimeoutEventByMask function.

Pass Conditions: All events should be disabled.

Fail Conditions: All other status

APIs called: ASF_GetSupportedTimeoutErrors, ASF_SetEventAsFatal, ASF_EnableTimeoutEventByMask, ASF_DisableTimeoutEventByMask

Covered Use Cases: Information about fault event, ASF Configuration

3.2.1.112. ASF_EnableTimoutEventByID

Test Description: Tests ASF_EnableTimeoutEventByID function.

Pass Conditions: All events should be enabled.

Fail Conditions: All other status

APIs called: ASF_GetSupportedTimeoutErrors, ASF_SetEventAsFatal, ASF_EnableTimeoutEventByID, ASF_DisableTimeoutEventByMask

Covered Use Cases: Information about fault event, ASF Configuration

3.2.1.113. ASF_DisableTimoutEventByID

Test Description: Tests ASF_DisableTimeoutEventByID function.

Pass Conditions: All events should be disabled.

Fail Conditions: All other status

APIs called: ASF_GetSupportedTimeoutErrors, ASF_SetEventAsFatal, ASF_EnableTimeoutEventByMask, ASF_DisableTimeoutEventByID

Covered Use Cases: Information about fault event, ASF Configuration

3.2.1.114. ASF_Statistic

Test Description: Tests ASF_ClearStatistic, ASF_RestoreStatistic

Pass Conditions: Value of counters should be as expected.

Fail Conditions: All other status

APIs called: ASF_EnableAllEvents, ASF_ClearStatistic, ASF_RestoreStatistic, ASF_GetStatistic, ASF_GetSupportedProtocolErrors, ASF_GetSupportedTimeoutErrors, ASF_EnableProtocolEventByMask, ASF_SetEventAsFatal

Covered Use Cases: ASF Events Statistic

3.2.1.115. ASF_testStopStartReinit

Test Description: Test purpose is to check if Start, Stop and Destroy functions work correctly.

Pass Conditions: Functions should work as expected.

Fail Conditions: All other status

APIs called: ASF_Stop, ASF_Start, ASF_Destroy

Covered Use Cases: Common architecture

3.2.1.116. ASF_testIncorrectDriverState

Test Description: In that test there is an attempt to use some API functions which should not be used when driver was not initialized properly. That is simulated through private data.

Pass Conditions: Functions should work as expected.

Fail Conditions: All other status

APIs called: all API functions

Covered Use Cases: Common architecture

3.2.2. Mechanical tests

In contrast to functional tests, mechanical tests are executed as user space application in RedHat 6.5 64bit. Hardware is emulated by memory based registers. There are three types of mechanical tests:

- Null pointer
- Range

- Set/get

During each generation of mechanical tests a random seed is used (and saved in the test log) to choose a set of 32-bit random numbers to reduce test runtime from trying all possible values.

3.2.2.1. Null pointer tests

These tests run through all pointer parameters of the API and call the function will all pointers properly initialized, but one null pointer and expect the function to return EINVAL in all cases. Note: there may be exceptions where a function would not need all pointers depending on values of other parameters, these can be covered in the YAML test exceptions.

3.2.2.2. Range tests

These tests call the function with a range (or a random subset) of values of all of the parameters and expect either positive or negative function return values depending on the parameter validity defined in that API definition YAML.

3.2.2.3. Set/Get tests

These tests call the function pairs responsible for controlling driver/controller operating parameters with a range (or a random subset) of input values of each parameter and expect positive function returns and matching parameter values to be returned by get as were provided to set.

Chapter 4. Test Results

All tests mentioned in this chapter are specific to the internal verification environment, and hence are not part of any release package.

4.1. Functional tests in RTL environment

4.1.1. Test Summary

For each test which runs, the result will be displayed as "PASSED" or "FAILED". Some tests may not be appropriate for some APIs, depending on configuration. If so these will be marked as "NOT SUPPORTED".

I3C Functional tests: 75 calls, 75 passed

ASF Functional tests: 18 calls, 18 passed

I3C Unit tests: 19 calls, 19 passed

4.1.2. Test Details

4.1.2.1. I3C Functional tests

```
test_sdr_master_sec_master_tx PASSED
test_sdr_master_sec_master_rx PASSED
test_sdr_master_sec_master_tx_threshold PASSED
test_sdr_master_sec_master_rx_threshold PASSED
test_sdr_master_sec_master_tx_multiple PASSED
test_sdr_master_slave_tx PASSED
test_sdr_master_slave_rx PASSED
test_ddr_master_sec_master_tx PASSED
test_ddr_master_sec_master_rx PASSED
test_ddr_master_sec_master_tx_threshold PASSED
test_ddr_master_sec_master_rx_threshold PASSED
test_ddr_master_slave_tx PASSED
test_ddr_master_slave_rx PASSED
test_sec_master_tx_no_data PASSED
test_sec_master_rx_without_daa PASSED
test_non_imm_bus_init PASSED
test_imm_bus_init PASSED
test_imm_non_imm_mix_bus_init PASSED
test_imm_in_progress PASSED
test_imm_inject PASSED
test_hot_join_sec_mster PASSED
test_hot_join_slave PASSED
test_ibi PASSED
test_ibi_nack PASSED
test_ibi_address_of_issuer PASSED
test_ibi_slot_overflow PASSED
test_ibi_tcam0_event PASSED
test_mastership_request_sec_mster PASSED
test_ccc_set_new_da PASSED
test_ccc_set_ad_from_sa_no_device PASSED
test_ccc_set_get_max_read_length PASSED
test_ccc_set_get_max_read_length_bcst_with_ibi PASSED
test_ccc_set_get_max_write_length PASSED
test_ccc_set_get_max_write_length_bcst PASSED
test_cmd_buffer_overflow PASSED
test_cmd_large_without_fifos PASSED
test_ccc_get_status PASSED
```

```
test_ccc_enter_test_mode PASSED
test_ccc_reset_daa_broadcast PASSED
test_ccc_enter_activity_state PASSED
test_ccc_set_get_max_data_speed PASSED
test_ccc_slave_event_invalid_mask PASSED
test_ccc_set_slave_events PASSED
test_ccc_set_nca_mode_in_priv_write PASSED
test_ccc_set_slave_buscon PASSED
test_ccc_set_target_reset PASSED
test_ccc_set_time_control PASSED
test_ccc_get_time_control PASSED
test_ccc_set_group_address PASSED
test_ccc_reset_group_address PASSED
test_init_params PASSED
test_init_sec_master_disable_interrupts PASSED
test_init_parts_core_enabled PASSED
test_init_stop_interrupt PASSED
test_init_destroy PASSED
test_init_bus_mode PASSED
test_init_without_devs PASSED
test_init_core_idle PASSED
test_init_too_many_devs PASSED
test_init_pure_bus PASSED
test_init_prescalers_with_no_timings PASSED
test_i2c_write PASSED
test_i2c_read PASSED
test_i2c_inactive_device_init PASSED
test_isr_no_interrupt PASSED
test_isr_tx_rx_thr_cmd_list_empty PASSED
test_auto_configuration PASSED
test_transfer_to_non_existing_device PASSED
test_get_asf_info PASSED
test_cmd_add_ddr_write PASSED
test_cmd_add_ddr_read PASSED
test_cmd_exec PASSED
test_get_slave_mrl_mwl PASSED
test_sdr_master_slave_fill_lvl PASSED
test_sdr_slave_flush_done PASSED
```

4.1.2.2. ASF Functional tests

```
ASF_testStopStartReinit PASSED
ASF_testIncorrectDriverState PASSED
ASF_testBIST1 PASSED
ASF_testBIST2 PASSED
ASF_testDisableAllEventsFunc PASSED
ASF_testEnableAllEventsFunc PASSED
ASF_testEnableEventFunc PASSED
ASF_testDisableEventFunc PASSED
ASF_testSetEventAsNonFatalFunc PASSED
ASF_testEnableProtocolEventByMaskFunc PASSED
ASF_testDisableProtocolEventByMaskFunc PASSED
ASF_testEnableProtocolEventByIDFunc PASSED
ASF_testDisableProtocolEventByIDFunc PASSED
ASF_testEnableTimeoutEventByMaskFunc PASSED
ASF_testDisableTimeoutEventByMaskFunc PASSED
ASF_testEnableTimeoutEventByIDFunc PASSED
ASF_testDisableTimeoutEventByIDFunc PASSED
ASF_testStatisticFunc PASSED
```

4.1.2.3. I3C Unit tests

```
test_cmd_list_init PASSED
test_cmd_list_max_fill PASSED
test_cmd_list_fill_and_empty_pop PASSED
test_cmd_list_try_to_overfill PASSED
test_cmd_list_try_to_overread PASSED
test_cmd_list_data_integrity_of_overfilled PASSED
test_cmd_list_search PASSED
test_cmd_list_delete_any PASSED
test_cmd_list_traverse PASSED
test_bytes_swap_3_bytes PASSED
test_bytes_swap_4_bytes PASSED
test_sanity_callbacks PASSED
test_sanity_function2 PASSED
test_sanity_function9 PASSED
test_sanity_function30 PASSED
test_sanity_function35 PASSED
test_sanity_function48 PASSED
test_sanity_function60 PASSED
test_sanity_function69 PASSED
```

4.2. Mechanical tests

4.2.1. Test Summary

Some tests may not be appropriate for some APIs, depending on configuration, if so these will be marked as "NOT SUPPORTED". For each test which runs, the result will be displayed as "PASSED" or "FAILED".

4.2.2. Test Details

Chapter 5. Static Analysis

5.1. Static Analysis summary

Static Analysis consists of 3 passes of tests against rule sets listed below:

Recommended Rules: 55 rules, 55 passed

MISRA C 2012: 322 rules, 300 passed, 5 waived, 17 failed

HIS Source Code Metrics: 11 rules, 5 passed, 2 waived, 4 failed

The details of Static Analysis results are present in various *.txt files present in the same folder (software/doc/test_reports). The results are present in below format:

Format for violation	Line number, rule, comment
Format for waived violation	Line number, file name, Parasoft's rule name (name of Parasoft's checker for a rule), Official rule name (name of rule in a specification).

5.2. Static Analysis

All Static Analysis was performed using DTP Engine for C/C++ 10.3.4 by Parasoft. For this process a 64-bit Linux environment was used. All violations are marked as FAILED in this report. Each waived rule is marked as WAIVED when a waiver was specified. For static analysis the following sets of rules were defined:

- Parasoft recommended rules
- MISRA C 2012
- HIS

There are three types of waivers:

Case-by-case	The software developer reviews violations reported for rules under this category and adds a comment to waive the issue if the reason provided in the "Waiver Reasoning" matches. Reasons other than those provided in the Reasoning are not accepted as waivers and the developer needs to fix the violation.
By files	Similar to "Case-by-case", but a single waiver covers all violations.
Permanently	The software developer ignores this rule. A global waiver fixes the violations.

A lists of the waived rules and directives can be found in the tables below. First table describes MISRA waivers, second - HIS waivers.

5.2.1. MISRA C 2012 waivers

Table 5.1. MISRA C 2012 waivers

Rule	Name	Type	Reasoning
Directive 4.1a	Avoid accessing arrays out of bounds	Case by case	CPPTest raises false violation. This case was discussed in ticket 00076766.
Directive 4.8	If a pointer to a structure or union is never dereferenced within a translation	Case by case	You can use direct access to functions or by object with pointers to functions. Because we provide library, we don't know, which model will be used, and we have to provide

Rule	Name	Type	Reasoning
	unit, then the implementation of the object should be hidden		all structures, even if not used in reference code.
Directive 4.9	A function should be used in preference to a function-like macro where they are interchangeable	By files	Specified in register access layer and for debug. The reason to use macros in hardware registers access functions is to concatenate the register name with the field name, SHIFT and MASK. Use macros in the debugging system so that you can build the driver without any debug strings in the object binary file.
Rule 1.1	The program shall contain no violations of the standard C syntax and constraints, and shall not exceed the implementations translation limits.	By files	Use this waiver to improve readability. The name length can be longer to keep more meaningful information. Our software naming convention requires prefix the module name before the Macro/function name. This results in long names.
Rule 2.7	There should be no unused parameters in functions	Case by case	Allows functions to ignore parameters to express more behaviors within a single type, which makes function pointers more useful.
Rule 3.1	The character sequences /* and // shall no be used within comment	Case by case	License body of third-party files can contain URL's. License body cannot be modified
Rule 5.4	Macro identifiers shall be distinct.	By files	The macro name reflects memory map structure. There are sets of macros for each field with the same name and appropriate postfix (SHIFT, WIDTH, MASK).
Rule 8.6	An identifier with external linkage shall have exactly one external definition.	By files	Definitions of those functions/variables must be provided by client/test. Those are platform dependent implementations.
Rule 8.7	Functions and objects should not be defined with external linkage if they are referenced in only one translation unit.	By files	These functions have to be visible because in complex driver's sanity functions are referenced from more than one translation unit.
Rule 8.13	A pointer parameter in a function prototype should be declared as pointer to cost if the pointer	Case by case	CPPTest raises false violation. This case was discussed in ticket 00060982. This waiver can also be used in cases where we define a function that has to be compatible with a prototype declared in external libraries, which we cannot modify.

Rule	Name	Type	Reasoning
	is not used to modify the addressed object.		
Rule 11.1	Conversions shall not be performed between a pointer to a function and any other type.	Case by case	CPPTest raises false violation. This case was discussed in ticket 00061812. This waiver could be used also if error injection is needed (to check error checking and correction, ECC).
Rule 11.3	A cast shall not be performed between a pointer to object type and a pointer to a different object type	Case by case	Access to object pointed by other object (like TRB fileds).
Rule 11.4	A conversion should not be performed between a pointer to object and an integer type	Case by case	Usually address to memory is stored as a pointer, but function writing to the registers take an integer argument (value). Also when base pointers have to be moved based on some configuration this can be used. This waiver is useful when register abstraction files don't use arrays.
Rule 12.2	The right-hand operand of a shift operator shall lie in the range zero to one less than the width in bits of the essential type of the left-hand operand.	By files	Shift values are generated by an internal tool and they will not exceed the "value" size. Shift range is verified on the generator side. Checking shift value here may cause performance issue.
Rule 16.1	All switch statements shall be well-formed	Case by case	Use this waiver to improve readability. There are fewer comparisions and jumps in C code, if we want to do something for each switch/ case.
Rule 17.1	The features of stdarg.h shall not be used.	By files	This waiver is applicable in firmware only, in files containing functions allowing transferring string messages with parameters and/or example implemetations of debug logging functions used in debug build.
Rule 18.1a	Avoid accessing arrays out of bounds	Case by case	CPPTest raises false violation. This case was discussed in ticket 00076766.
Rule 20.9	All identifiers used in the controlling expression of #if or ##elif preprocessing directives shall be #define'd before evaluation.	Case by case	CPPTest raises false violation. This case was discussed in ticket 00062046.
Rule 20.10	The # and ## preprocessor	By files	The places in which order of evaluation has no side effects. Used also in hardware

Rule	Name	Type	Reasoning
	operators should not be used		registers access macros to use field SHIFT and MASK value from the field name.
Rule 21.6	The Standard Library input/output functions shall not be used.	By files	This waiver is applicable in firmware only, in files containing functions allowing transferring string messages with parameters and/or example implemetations of debug logging functions used in debug build.

5.2.2. HIS waivers

Table 5.2. HIS waivers

Rule	Name	Type	Reasoning
Metric CALLING	A function should not be called from more than 5 different functions	Permanently	There are basic library functions which are used, for example, for hardware registers access or for logging. Those functions are called many times in the driver. Trying to apply this rule might cause the code to be come completely incomprehensible.
Metric VOCF	The language scope is an indicator of the cost of maintaining/ changing functions.	By files	Sanity functions require many parameter checks. This checking causes an increase of VOCF because each check is done in a similar way.
Metric v(G)	McCabe Cyclomatic Complexity	By files	An auto-generated sanity code for checking a large structure is more readable, when all fields are checked in one function.
Metric COMF	Relationship of the number of comments (outside of and within functions) to the number of statements > 0.2.	By files	Auto-generated code is written in self-commented way and it doesn't require extra comments.
Metric CALLS	A function should not call more than 7 different functions	Case by case, applicable in firmware code only	This waiver is used to avoid unclear and artificially split code in main() function.

5.2.3. Other waivers

Table 5.3. Other waivers

Rule	Name	Type	Reasoning
Avoid endless loops	A loop should have termination condition	Case by case	For testing code it's useful to have functions that never ends to mark test complete or when the processor reaches a particular point in code.
Avoid code duplication	A code in a project should not be duplicated	Case by case	Functional tests are themselves examples, showing how some IP functionality should be performed by software. One, lenghty

Rule	Name	Type	Reasoning
			function showing one use case is much simpler to analyze.

5.2.4. Parasoft DTP Engine for C/C++ Analysis - Recommended Rules

```
Do not pass negative values to functions expecting non-negative arguments
     (BD-API-NEGPARAM) PASSED
Always catch exceptions (BD-PB-EXCEPT) PASSED
Avoid use before initialization (BD-PB-NOTINIT) PASSED
Avoid null pointer dereferencing (BD-PB-NP) PASSED
Avoid buffer overflow due to defining incorrect format limits
     (BD-PB-OVERFFMT) PASSED
Avoid overflow due to reading a not zero terminated string (BD-PB-OVERFNZT)
     PASSED
Avoid overflow when reading from a buffer (BD-PB-OVERFRD) PASSED
Avoid overflow when writing to a buffer (BD-PB-OVERFWR) PASSED
Avoid division by zero (BD-PB-ZERO) PASSED
Avoid accessing arrays out of bounds (BD-PB-ARRAY) PASSED
Avoid conditions that always evaluate to the same value (BD-PB-CC) PASSED
Do not check for null after dereferencing (BD-PB-DEREF) PASSED
Suspicious setting of stream flags (BD-PB-STREAMFLAGS) PASSED
Restore stream format (BD-PB-STREAMFMT) PASSED
Properly deallocate dynamically allocated resources (BD-RES-BADDEALLOC)
     PASSED
Do not use resources that have been freed (BD-RES-FREE) PASSED
Do not free resources using invalid pointers (BD-RES-INVFREE) PASSED
Ensure resources are freed (BD-RES-LEAKS) PASSED
Avoid double locking (BD-TRS-DLOCK) PASSED
Avoid race conditions when using fork and file descriptors (BD-TRS-FORKFILE)
     PASSED
Do not abandon unreleased locks (BD-TRS-LOCK) PASSED
Do not acquire locks in different order (BD-TRS-ORDER) PASSED
Avoid race conditions while checking for the existence of a symbolic link
     (BD-TRS-SYMLINK) PASSED
Do not use blocking functions while holding a lock (BD-TRS-TSHL) PASSED
Avoid function duplication (CDD-DUPM) PASSED
Local variables should not use the same names as member variables
     (CODSTA-44) PASSED
Pointer should not be compared with NULL using relational operators <, >,
     >=, <= (CODSTA-147) PASSED
Do not use string literals as operands of equality or relational operators
     (CODSTA-148) PASSED
Avoid infinite loops (CODSTA-82) PASSED
Constructors allowing for conversion should be made explicit (CODSTA-CPP-04)
     PASSED
Throw by value, catch by reference (EXCEPT-02) PASSED
Do not throw from within destructor (EXCEPT-03) PASSED
All member variables should be initialized in constructor (INIT-06) PASSED
McCabe Cyclomatic Complexity (METRIC.CC) PASSED
Nested Blocks Depth (METRIC.NBD) PASSED
Floating-point expressions shall not be tested for equality or inequality
     (MISRA2004-13_3) PASSED
All exit paths from a function with non-void return type shall have an
     explicit return statement with an expression (MISRA2004-16_8) PASSED
The address of an object with automatic storage shall not be returned from a
     function (MISRA2004-17_6_a) PASSED
Do not invoke malloc/realloc for objects having constructors (MRM-08) PASSED
Declare a copy assignment operator for classes with dynamically allocated
     memory (MRM-37) PASSED
Declare a copy constructor for classes with dynamically allocated memory
     (MRM-38) PASSED
```

```
Never provide brackets ([]) for delete when deallocating non-arrays (MRM-35)
     PASSED
Always provide empty brackets ([]) for delete when deallocating arrays
     (MRM-36) PASSED
Do not use 'delete' on pointers to a void type (MRM-51) PASSED
Class cannot inherit other class more than once unless it is virtual
     inheritance (OOP-03) PASSED
Avoid calling virtual functions from constructors (OOP-16) PASSED
If a class has virtual functions it shall have a virtual destructor (OOP-23)
     PASSED
Pass objects by reference instead of by value (OPT-14) PASSED
Do not call delete on non-pointers (PB-13) PASSED
Properly terminate character strings (PB-21) PASSED
Do not cast from or to incomplete class at the point of casting (PB-54)
     PASSED
Do not delete objects with incomplete class at the point of deletion (PB-55)
     PASSED
Boolean condition always evaluates to the same value due to enumeration with
     only zero or only non-zero constants (PB-68) PASSED
Suspicious argument to malloc (PB-60) PASSED
Pointer arithmetic performed on freshly allocated memory (PB-61) PASSED
```

5.2.5. Parasoft DTP Engine for C/C++ Analysis - MISRA C 2012

```
Avoid accessing arrays out of bounds (MISRAC2012-DIR_4_1-a) PASSED
Avoid null pointer dereferencing (MISRAC2012-DIR_4_1-b) PASSED
Avoid division by zero (MISRAC2012-DIR_4_1-c) PASSED
Avoid buffer overflow due to defining incorrect format limits
     (MISRAC2012-DIR_4_1-d) PASSED
Avoid overflow due to reading a not zero terminated string
     (MISRAC2012-DIR_4_1-e) PASSED
Do not check for null after dereferencing (MISRAC2012-DIR_4_1-f) PASSED
Avoid overflow when reading from a buffer (MISRAC2012-DIR_4_1-g) PASSED
Avoid overflow when writing to a buffer (MISRAC2012-DIR_4_1-h) PASSED
Do not subtract two pointers that do not address elements of the same array
     (MISRAC2012-DIR_4_1-i) PASSED
Do not compare two unrelated pointers (MISRAC2012-DIR_4_1-j) PASSED
Avoid integer overflows (MISRAC2012-DIR_4_1-k) FAILED
Use multiple include guards (MISRAC2012-DIR_4_10-a) PASSED
Validate values passed to library functions (MISRAC2012-DIR_4_11-a) PASSED
Dynamic heap memory allocation shall not be used (MISRAC2012-DIR_4_12-a)
     PASSED
Ensure resources are freed (MISRAC2012-DIR_4_13-a) PASSED
Do not use resources that have been freed (MISRAC2012-DIR_4_13-b) PASSED
Do not free resources using invalid pointers (MISRAC2012-DIR_4_13-c) PASSED
Do not abandon unreleased locks (MISRAC2012-DIR_4_13-d) PASSED
Avoid double locking (MISRAC2012-DIR_4_13-e) PASSED
Do not release a lock that has not been acquired (MISRAC2012-DIR_4_13-f)
     PASSED
Avoid tainted data in array indexes (MISRAC2012-DIR_4_14-a) PASSED
Protect against integer overflow/underflow from tainted data
     (MISRAC2012-DIR_4_14-b) PASSED
Avoid buffer read overflow from tainted data (MISRAC2012-DIR_4_14-c) PASSED
Avoid buffer write overflow from tainted data (MISRAC2012-DIR_4_14-d) PASSED
Protect against command injection (MISRAC2012-DIR_4_14-e) PASSED
Protect against file name injection (MISRAC2012-DIR_4_14-f) PASSED
Protect against SQL injection (MISRAC2012-DIR_4_14-g) PASSED
Prevent buffer overflows from tainted data (MISRAC2012-DIR_4_14-h) PASSED
Avoid buffer overflow from tainted data due to defining incorrect format
     limits (MISRAC2012-DIR_4_14-i) PASSED
Protect against environment injection (MISRAC2012-DIR_4_14-j) PASSED
Avoid printing tainted data on the output console (MISRAC2012-DIR_4_14-k)
     PASSED
```

```
Exclude unsanitized user input from format strings (MISRAC2012-DIR_4_14-l)
     PASSED
All usage of assembler shall be documented (MISRAC2012-DIR_4_2-a) PASSED
Assembly language shall be encapsulated and isolated (MISRAC2012-DIR_4_3-a)
     PASSED
Sections of code should not be "commented out"
     (MISRAC2012-DIR_4_4-a) FAILED
Identifiers in the same name space with overlapping visibility should be
     typographically unambiguous (MISRAC2012-DIR_4_5-a) PASSED
typedefs to basic types should contain some digits in their name
     (MISRAC2012-DIR_4_6-a) PASSED
typedefs should be used in place of the basic types (MISRAC2012-DIR_4_6-b)
     PASSED
Use typedefs from stdint.h instead of declaring your own in C99 code
     (MISRAC2012-DIR_4_6-c) PASSED
Consistently check the returned value of non-void functions
     (MISRAC2012-DIR_4_7-a) PASSED
Always check the returned value of non-void function (MISRAC2012-DIR_4_7-b)
     PASSED
If a pointer to a structure or union is never dereferenced within a
     translation unit, then the implementation of the object should be
     hidden (MISRAC2012-DIR_4_8-a) FAILED
A function should be used in preference to a function-like macro
     (MISRAC2012-DIR_4_9-a) WAIVED
An expression of essentially Boolean type should always be used where an
     operand is interpreted as a Boolean value (MISRAC2012-RULE_10_1-a)
     FAILED
An operand of essentially Boolean type should not be used where an operand
     is interpreted as a numeric value (MISRAC2012-RULE_10_1-b) PASSED
An operand of essentially character type should not be used where an operand
     is interpreted as a numeric value (MISRAC2012-RULE_10_1-c) PASSED
An operand of essentially enum type should not be used in an arithmetic
     operation (MISRAC2012-RULE_10_1-d) PASSED
Shift and bitwise operations should not be performed on operands of
     essentially signed or enum type (MISRAC2012-RULE_10_1-e) PASSED
An operand of essentially signed or enum type should not be used as the
     right hand operand to the bitwise shifting operator
     (MISRAC2012-RULE_10_1-f) PASSED
An operand of essentially unsigned type should not be used as the operand to
     the unary minus operator (MISRAC2012-RULE_10_1-g) PASSED
Expressions of essentially character type shall not be used inappropriately
     in addition and subtraction operations (MISRAC2012-RULE_10_2-a) PASSED
The value of an expression shall not be assigned to an object with a
     narrower essential type (MISRAC2012-RULE_10_3-a) PASSED
The value of an expression shall not be assigned to an object of a different
     essential type category (MISRAC2012-RULE_10_3-b) PASSED
Both operands of an operator in which the usual arithmetic conversions are
     performed shall have the same essential type category
     (MISRAC2012-RULE_10_4-a) PASSED
The second and third operands of the ternary operator shall have the same
     essential type category (MISRAC2012-RULE_10_4-b) PASSED
The cast operation to essentially enumeration type is not allowed
     (MISRAC2012-RULE_10_5-a) PASSED
Do not cast from or to essentially Boolean type (MISRAC2012-RULE_10_5-b)
     PASSED
Do not use casts between essentially character types and essentially
     floating types (MISRAC2012-RULE_10_5-c) PASSED
The value of a composite expression shall not be assigned to an object with
     wider essential type (MISRAC2012-RULE_10_6-a) FAILED
If a composite expression is used as one operand of an operator in which the
     usual arithmetic conversions are performed then the other operand shall
     not have wider essential type (MISRAC2012-RULE_10_7-a) PASSED
If a composite expression is used as one (second or third) operand of a
     conditional operator then the other operand shall not have wider
```

```
     essential type (MISRAC2012-RULE_10_7-b) PASSED
The value of a composite expression shall not be cast to a different
     essential type category or a wider essential type
     (MISRAC2012-RULE_10_8-a) FAILED
Conversions shall not be performed between a pointer to a function and any
     other type than pointer to function (MISRAC2012-RULE_11_1-a) PASSED
Conversions shall not be performed between non compatible pointer to a
     function types (MISRAC2012-RULE_11_1-b) PASSED
Conversions shall not be performed between a pointer to an incomplete type
     and any other type (MISRAC2012-RULE_11_2-a) PASSED
A cast shall not be performed between a pointer to object type and a pointer
     to a different object type (MISRAC2012-RULE_11_3-a) PASSED
A conversion should not be performed between a pointer to object and an
     integer type (MISRAC2012-RULE_11_4-a) PASSED
A conversion should not be performed from pointer to void into pointer to
     object (MISRAC2012-RULE_11_5-a) PASSED
A cast shall not be performed between pointer to void and an arithmetic type
     (MISRAC2012-RULE_11_6-a) PASSED
A cast shall not be performed between pointer to object and a non-integer
     arithmetic type (MISRAC2012-RULE_11_7-a) PASSED
A cast shall not remove any 'const' or 'volatile' qualification from the
     type of a pointer or reference (MISRAC2012-RULE_11_8-a) PASSED
Literal zero (0) shall not be used as the null-pointer-constant
     (MISRAC2012-RULE_11_9-a) PASSED
Use NULL instead of literal zero (0) as the null-pointer-constant
     (MISRAC2012-RULE_11_9-b) PASSED
Use parentheses unless all operators in the expression are the same
     (MISRAC2012-RULE_12_1-a) PASSED
The operands of a logical && or || shall be primary-expressions
     (MISRAC2012-RULE_12_1-b) PASSED
The operand of the 'sizeof' operator should be enclosed in parentheses
     (MISRAC2012-RULE_12_1-c) PASSED
The right-hand operand of a shift operator shall lie between zero and one
     less than the width in bits of the underlying type of the left-hand
     operand (MISRAC2012-RULE_12_2-a) WAIVED
The comma operator shall not be used (MISRAC2012-RULE_12_3-a) PASSED
Integer overflow or underflow in constant expression in '+', '-', '*'
     operator (MISRAC2012-RULE_12_4-a) PASSED
Integer overflow or underflow in constant expression in '<<' operator
     (MISRAC2012-RULE_12_4-b) PASSED
The 'sizeof' operator shall not have an operand which is a function
     parameter declared as "array of type"
     (MISRAC2012-RULE_12_5-a) PASSED
Initializer lists shall not contain persistent side effects
     (MISRAC2012-RULE_13_1-a) PASSED
The value of an expression shall be the same under any order of evaluation
     that the standard permits (MISRAC2012-RULE_13_2-a) PASSED
Don't write code that depends on the order of evaluation of function
     arguments (MISRAC2012-RULE_13_2-b) PASSED
Don't write code that depends on the order of evaluation of function
     designator and function arguments (MISRAC2012-RULE_13_2-c) PASSED
Don't write code that depends on the order of evaluation of expression that
     involves a function call (MISRAC2012-RULE_13_2-d) PASSED
Between sequence points an object shall have its stored value modified at
     most once by the evaluation of an expression (MISRAC2012-RULE_13_2-e)
     PASSED
Do not use more than one volatile between two adjacent sequence points
     (MISRAC2012-RULE_13_2-f) PASSED
Don't write code that depends on the order of evaluation of function calls
     (MISRAC2012-RULE_13_2-g) PASSED
A full expression containing an increment (++) or decrement (--) operator
     should have no other potential side effects (MISRAC2012-RULE_13_3-a)
     PASSED
The result of a built-in assignment operator should not be used
```

```
     (MISRAC2012-RULE_13_4-a) PASSED
The right-hand operand of a logical && or || operator shall not
     contain side effects (MISRAC2012-RULE_13_5-a) FAILED
The operand of the sizeof operator shall not contain any expression which
     has side effects (MISRAC2012-RULE_13_6-a) PASSED
Object designated by a volatile lvalue should not be accessed in the operand
     of the sizeof operator (MISRAC2012-RULE_13_6-b) PASSED
The function call shall not be the operand of the sizeof operator
     (MISRAC2012-RULE_13_6-c) PASSED
A loop counter in a 'for' loop shall not have essentially floating type
     (MISRAC2012-RULE_14_1-a) PASSED
A loop counter in 'while' and 'do-while' loops shall not have essentially
     floating type (MISRAC2012-RULE_14_1-b) PASSED
There shall only be one loop counter in a 'for' loop, which shall not be
     modified in the 'for' loop body (MISRAC2012-RULE_14_2-a) PASSED
The first clause of a 'for' loop shall be well-formed
     (MISRAC2012-RULE_14_2-b) PASSED
The second clause of a 'for' loop shall be well-formed
     (MISRAC2012-RULE_14_2-c) PASSED
The third clause of a 'for' statement shall be well-formed
     (MISRAC2012-RULE_14_2-d) PASSED
Avoid conditions that always evaluate to the same value
     (MISRAC2012-RULE_14_3-ac) WAIVED
Tests of a value against zero should be made explicit, unless the operand is
     effectively Boolean (MISRAC2012-RULE_14_4-a) FAILED
The goto statement shall not be used (MISRAC2012-RULE_15_1-a) PASSED
The goto statement shall jump to a label declared later in the same function
     body (MISRAC2012-RULE_15_2-a) PASSED
Any label referenced by a goto statement shall be declared in the same
     block, or in a block enclosing the goto statement
     (MISRAC2012-RULE_15_3-a) PASSED
For any iteration statement there shall be no more than one break or goto
     statement used for loop termination (MISRAC2012-RULE_15_4-a) PASSED
A function shall have a single point of exit at the end of the function
     (MISRAC2012-RULE_15_5-a) PASSED
The statement forming the body of a 'switch', 'while', 'do...while' or 'for'
     statement shall be a compound statement (MISRAC2012-RULE_15_6-a) PASSED
'if' and 'else' should be followed by a compound statement
     (MISRAC2012-RULE_15_6-b) PASSED
All 'if...else-if' constructs shall be terminated with an 'else' clause
     (MISRAC2012-RULE_15_7-a) PASSED
A switch statement shall only contain switch labels and switch clauses, and
     no other code (MISRAC2012-RULE_16_1-a) PASSED
A switch label shall only be used when the most closely-enclosing compound
     statement is the body of a switch statement (MISRAC2012-RULE_16_1-b)
     PASSED
An unconditional break statement shall terminate every non-empty case clause
     (MISRAC2012-RULE_16_1-c) PASSED
An unconditional break statement shall terminate every non-empty default
     clause (MISRAC2012-RULE_16_1-d) PASSED
Always provide a default branch for switch statements
     (MISRAC2012-RULE_16_1-e) PASSED
A 'default' label shall have a statement or a comment before terminating
     'break' (MISRAC2012-RULE_16_1-f) PASSED
A 'default' label, if it exists, shall appear as either the first or the
     last switch label of a switch statement (MISRAC2012-RULE_16_1-g) PASSED
Every switch statement shall have at least two switch-clauses
     (MISRAC2012-RULE_16_1-h) PASSED
A switch label shall only be used when the most closely-enclosing compound
     statement is the body of a switch statement (MISRAC2012-RULE_16_2-a)
     PASSED
An unconditional break statement shall terminate every non-empty case clause
     (MISRAC2012-RULE_16_3-a) PASSED
An unconditional break statement shall terminate every non-empty default
```

```
     clause (MISRAC2012-RULE_16_3-b) PASSED
Always provide a default branch for switch statements
     (MISRAC2012-RULE_16_4-a) PASSED
A 'default' label shall have a statement or a comment before terminating
     'break' (MISRAC2012-RULE_16_4-b) PASSED
A 'default' label, if it exists, shall appear as either the first or the
     last switch label of a switch statement (MISRAC2012-RULE_16_5-a) PASSED
Every switch statement shall have at least two switch-clauses
     (MISRAC2012-RULE_16_6-a) PASSED
A switch expression shall not represent a value that is effectively Boolean
     (MISRAC2012-RULE_16_7-a) PASSED
A switch expression shall not represent a value that is effectively Boolean
     (MISRAC2012-RULE_16_7-b) PASSED
The identifiers va_list, va_arg, va_start, va_end, va_copy should not be
     used (MISRAC2012-RULE_17_1-a) PASSED
The identifiers va_list, va_arg, va_start, va_end should not be used
     (MISRAC2012-RULE_17_1-b) PASSED
Functions shall not call themselves, either directly or indirectly
     (MISRAC2012-RULE_17_2-a) PASSED
Functions shall always have visible prototype at the function call
     (MISRAC2012-RULE_17_3-a) PASSED
All exit paths from a function with non-void return type shall have an
     explicit return statement with an expression (MISRAC2012-RULE_17_4-a)
     PASSED
All exit paths from a function, except main(), with non-void return type
     shall have an explicit return statement with an expression
     (MISRAC2012-RULE_17_4-b) PASSED
The function argument corresponding to a parameter declared to have an array
     type shall have an appropriate number of elements
     (MISRAC2012-RULE_17_5-a) PASSED
The declaration of an array parameter shall not contain the 'static' keyword
     between the [ ] (MISRAC2012-RULE_17_6-a) PASSED
The value returned by a function having non-void return type shall be used
     (MISRAC2012-RULE_17_7-a) PASSED
The value returned by a function having non-void return type shall be used
     (MISRAC2012-RULE_17_7-b) PASSED
A function parameter should not be modified (MISRAC2012-RULE_17_8-a) PASSED
Avoid accessing arrays out of bounds (MISRAC2012-RULE_18_1-a) PASSED
Avoid accessing arrays and pointers out of bounds (MISRAC2012-RULE_18_1-b)
     PASSED
A pointer operand and any pointer resulting from pointer arithmetic using
     that operand shall both address elements of the same array
     (MISRAC2012-RULE_18_1-c) PASSED
Do not subtract two pointers that do not address elements of the same array
     (MISRAC2012-RULE_18_2-a) PASSED
Do not compare two unrelated pointers (MISRAC2012-RULE_18_3-a) PASSED
The +, -, += and -= operators should not be applied to an expression of
     pointer type (MISRAC2012-RULE_18_4-a) PASSED
The declaration of objects should contain no more than 2 levels of pointer
     indirection (MISRAC2012-RULE_18_5-a) PASSED
The address of an object with automatic storage shall not be returned from a
     function (MISRAC2012-RULE_18_6-a) PASSED
The address of an object with automatic storage shall not be assigned to
     another object that may persist after the first object has ceased to
     exist (MISRAC2012-RULE_18_6-b) PASSED
Flexible array members shall not be declared (MISRAC2012-RULE_18_7-a) PASSED
Variable-length array types shall not be used (MISRAC2012-RULE_18_8-a)
     PASSED
An object shall not be assigned to an overlapping object
     (MISRAC2012-RULE_19_1-a) PASSED
An object shall not be assigned to an overlapping object
     (MISRAC2012-RULE_19_1-b) PASSED
An object shall not be assigned or copied to an overlapping object
     (MISRAC2012-RULE_19_1-c) PASSED
```

```
The union keyword should not be used (MISRAC2012-RULE_19_2-a) PASSED
A program should not exceed the translation limits imposed by The Standard
     (c90) (MISRAC2012-RULE_1_1-a) FAILED
A program should not exceed the translation limits imposed by The Standard
     (c99) (MISRAC2012-RULE_1_1-b) PASSED
A program should not exceed the translation limits imposed by The Standard
     (c90) (MISRAC2012-RULE_1_1-c) FAILED
A program should not exceed the translation limits imposed by The Standard
     (c99) (MISRAC2012-RULE_1_1-d) FAILED
Avoid division by zero (MISRAC2012-RULE_1_3-a) PASSED
Avoid use before initialization (MISRAC2012-RULE_1_3-b) PASSED
Do not use resources that have been freed (MISRAC2012-RULE_1_3-c) PASSED
Avoid overflow when reading from a buffer (MISRAC2012-RULE_1_3-d) PASSED
Avoid overflow when writing to a buffer (MISRAC2012-RULE_1_3-e) PASSED
The value of an expression shall be the same under any order of evaluation
     that the standard permits (MISRAC2012-RULE_1_3-f) PASSED
Don't write code that depends on the order of evaluation of function
     arguments (MISRAC2012-RULE_1_3-g) PASSED
Don't write code that depends on the order of evaluation of function
     designator and function arguments (MISRAC2012-RULE_1_3-h) PASSED
Don't write code that depends on the order of evaluation of expression that
     involves a function call (MISRAC2012-RULE_1_3-i) PASSED
Between sequence points an object shall have its stored value modified at
     most once by the evaluation of an expression (MISRAC2012-RULE_1_3-j)
     PASSED
Do not use more than one volatile between two adjacent sequence points
     (MISRAC2012-RULE_1_3-k) PASSED
Don't write code that depends on the order of evaluation of function calls
     (MISRAC2012-RULE_1_3-l) PASSED
The address of an object with automatic storage shall not be returned from a
     function (MISRAC2012-RULE_1_3-m) PASSED
The address of an object with automatic storage shall not be assigned to
     another object that may persist after the first object has ceased to
     exist (MISRAC2012-RULE_1_3-n) PASSED
The left-hand operand of a right-shift operator shall not have a negative
     value (MISRAC2012-RULE_1_3-o) PASSED
The '_Generic' operator should not be used (MISRAC2012-RULE_1_4-a) PASSED
The '_Noreturn' function specifier should not be used
     (MISRAC2012-RULE_1_4-b) PASSED
The <stdnoreturn.h> header file should not be used
     (MISRAC2012-RULE_1_4-c) PASSED
The '_Atomic' type specifier and the '_Atomic' type qualifier should not be
     used (MISRAC2012-RULE_1_4-d) PASSED
The facilities that are specified as being provided by <stdatomic.h>
     should not be used (MISRAC2012-RULE_1_4-e) PASSED
The '_Thread_local' storage class specifier should not be used
     (MISRAC2012-RULE_1_4-f) PASSED
The facilities that are specified as being provided by <threads.h> should
     not be used (MISRAC2012-RULE_1_4-g) PASSED
The '_Alignas' alignment specifier and the '_Alignof' operator should not be
     used (MISRAC2012-RULE_1_4-h) PASSED
The <stdalign.h> header file shall not be used (MISRAC2012-RULE_1_4-i)
     PASSED
The '__STDC_WANT_LIB_EXT1__' macro should not be defined to the value other
     than '0' (MISRAC2012-RULE_1_4-j) PASSED
The 'rsize_t' type should not be used (MISRAC2012-RULE_1_4-k) PASSED
The 'errno_t' type should not be used (MISRAC2012-RULE_1_4-l) PASSED
Do not use following macros: RSIZE_MAX, L_tmpnam_s, TMP_MAX_S
     (MISRAC2012-RULE_1_4-m) PASSED
Do not use the functions defined in Annex K of ISO/IEC 9899:2011 standard
     (MISRAC2012-RULE_1_4-n) PASSED
#include statements in a file should only be preceded by other preprocessor
     directives or comments (MISRAC2012-RULE_20_1-a) PASSED
The # and ## preprocessor operators should not be used
```

```
     (MISRAC2012-RULE_20_10-a) WAIVED
A macro parameter immediately following a # operator shall not immediately
     be followed by or preceded by a ## operator (MISRAC2012-RULE_20_11-a)
     PASSED
A macro parameter used as an operand to the # or ## operators, which is
     itself subject to further macro replacement, shall only be used as an
     operand to these operators (MISRAC2012-RULE_20_12-a) PASSED
Preprocessing directives shall be syntactically meaningful even when
     excluded by the preprocessor (MISRAC2012-RULE_20_13-a) PASSED
All #else, #elif and #endif preprocessor directives shall reside in the same
     file as the #if or #ifdef directive to which they are related
     (MISRAC2012-RULE_20_14-a) PASSED
The ', ", /0-DISABLE_NOUVEAU.TXT /apps /backup-ldapmatch /bin /boot
     /cds /cgroup /depot /dev /eng /etc /file /grid /hls /home /icd /lan
     /lib /lib64 /local /lost+found /media /misc /mnt /net /newdepot /opt
     /proc /process /project /projects /root /rscratch /sbin /scratch
     /selinux /servers /srv /sys /test /tmp /userspace /usr /var /vols or //
     characters shall not occur in a header file name
     (MISRAC2012-RULE_20_2-a) PASSED
The character should not occur in a header file name
     (MISRAC2012-RULE_20_2-b) PASSED
The #include directive shall be followed by either a <filename> or
     "filename" sequence (MISRAC2012-RULE_20_3-a) PASSED
A macro shall not be defined with the same name as a keyword in C90
     (MISRAC2012-RULE_20_4-a) PASSED
A macro shall not be defined with the same name as a keyword in C99
     (MISRAC2012-RULE_20_4-b) PASSED
#undef shall not be used (MISRAC2012-RULE_20_5-a) PASSED
Arguments to a function-like macro shall not contain tokens that look like
     preprocessing directives (MISRAC2012-RULE_20_6-a) PASSED
In the definition of a function-like macro each instance of a parameter
     shall be enclosed in parentheses unless it is used as the operand of #
     or ## (MISRAC2012-RULE_20_7-a) PASSED
The controlling expression of a #if or #elif preprocessing directive shall
     evaluate to 0 or 1 (MISRAC2012-RULE_20_8-a) PASSED
Do not use in preprocessor directives #if and #elif macros not defined in
     translation unit (MISRAC2012-RULE_20_9-b) PASSED
Do not #define or #undef identifiers with names which start with underscore
     (MISRAC2012-RULE_21_1-a) PASSED
Reserved identifiers, macros and functions in the standard library, shall
     not be defined, redefined or undefined (C90 code)
     (MISRAC2012-RULE_21_1-b) PASSED
Reserved identifiers, macros and functions in the standard library, shall
     not be defined, redefined or undefined (C99 code)
     (MISRAC2012-RULE_21_1-c) PASSED
Do not #define nor #undef identifier 'defined' (MISRAC2012-RULE_21_1-d)
     PASSED
Avoid functions which use time from standard C library
     (MISRAC2012-RULE_21_10-a) PASSED
The standard header file <tgmath.h> shall not be used
     (MISRAC2012-RULE_21_11-a) PASSED
The facilities that are specified as being provided by <tgmath.h> should
     not be used (MISRAC2012-RULE_21_11-b) PASSED
The exception handling features of <fenv.h> should not be used
     (MISRAC2012-RULE_21_12-a) PASSED
Do not pass incorrect values to ctype.h library functions
     (MISRAC2012-RULE_21_13-a) PASSED
The Standard Library function memcmp shall not be used to compare null
     terminated strings (MISRAC2012-RULE_21_14-a) PASSED
The pointer arguments to the Standard Library functions 'memcmp', 'memmove'
     and 'memcmp' shall be pointers to qualified or unqualified versions of
     compatible types (MISRAC2012-RULE_21_15-a) PASSED
The pointer arguments to the Standard Library function 'memcmp' shall point
     to either a pointer type, an essentially signed type, an essentially
```

```
     unsigned type, an essentially Boolean type or an essentially enum type
     (MISRAC2012-RULE_21_16-a) PASSED
Avoid overflow due to reading a not zero terminated string
     (MISRAC2012-RULE_21_17-a) PASSED
Avoid overflow when writing to a buffer (MISRAC2012-RULE_21_17-b) PASSED
The size_t argument passed to any function in string.h shall have an
     appropriate value (MISRAC2012-RULE_21_18-a) PASSED
The pointers returned by the Standard Library functions 'localeconv',
     'getenv', 'setlocale' or, 'strerror' shall only be used as if they have
     pointer to const-qualified type (MISRAC2012-RULE_21_19-a) PASSED
Strings pointed by members of the structure 'lconv' should not be modified
     (MISRAC2012-RULE_21_19-b) PASSED
The names of standard library macros, objects and functions shall not be
     reused (MISRAC2012-RULE_21_2-a) PASSED
The names of standard library macros, objects and functions shall not be
     reused (C90) (MISRAC2012-RULE_21_2-b) PASSED
The names of standard library macros, objects and functions shall not be
     reused (C99) (MISRAC2012-RULE_21_2-c) PASSED
Pointers returned by certain Standard Library functions should not be used
     following a subsequent call to the same or related function
     (MISRAC2012-RULE_21_20-a) PASSED
The 'system()' function from the 'stdlib.h' or 'cstdlib' library shall not
     be used (MISRAC2012-RULE_21_21-a) PASSED
Dynamic heap memory allocation shall not be used (MISRAC2012-RULE_21_3-a)
     PASSED
The setjmp macro and the longjmp function shall not be used
     (MISRAC2012-RULE_21_4-a) PASSED
The standard header file <setjmp.h> shall not be used
     (MISRAC2012-RULE_21_4-b) PASSED
The standard header file <signal.h> shall not be used
     (MISRAC2012-RULE_21_5-a) PASSED
The signal handling facilities of <signal.h> shall not be used
     (MISRAC2012-RULE_21_5-b) PASSED
The Standard Library input/output functions shall not be used
     (MISRAC2012-RULE_21_6-a) PASSED
The library functions atof, atoi and atol from library stdlib.h shall not be
     used (MISRAC2012-RULE_21_7-a) PASSED
The 'abort()' function from the 'stdlib.h' or 'cstdlib' library shall not be
     used (MISRAC2012-RULE_21_8-a) PASSED
The 'exit()' function from the 'stdlib.h' or 'cstdlib' library shall not be
     used (MISRAC2012-RULE_21_8-b) PASSED
The 'quick_exit()' and '_Exit()' functions from the 'stdlib.h' or 'cstdlib'
     library shall not be used (MISRAC2012-RULE_21_8-c) PASSED
The library functions bsearch and qsort of <stdlib.h> shall not be used
     (MISRAC2012-RULE_21_9-a) PASSED
Ensure resources are freed (MISRAC2012-RULE_22_1-a) PASSED
Properly use errno value (MISRAC2012-RULE_22_10-a) PASSED
Do not use resources that have been freed (MISRAC2012-RULE_22_2-a) PASSED
Do not free resources using invalid pointers (MISRAC2012-RULE_22_2-b) PASSED
The same file shall not be opened for read and write access at the same time
     on different streams (MISRAC2012-RULE_22_3-a) PASSED
Avoid writing to a stream which has been opened as read only
     (MISRAC2012-RULE_22_4-a) PASSED
A pointer to a FILE object shall not be dereferenced
     (MISRAC2012-RULE_22_5-a) PASSED
A pointer to a FILE object shall not be dereferenced by a library function
     (MISRAC2012-RULE_22_5-b) PASSED
Do not use resources that have been freed (MISRAC2012-RULE_22_6-a) PASSED
The macro EOF should be compared with the unmodified return value from the
     Standard Library function (MISRAC2012-RULE_22_7-a) PASSED
Properly use errno value (MISRAC2012-RULE_22_8-a) PASSED
Properly use errno value (MISRAC2012-RULE_22_9-a) PASSED
There shall be no unreachable code in "else" block
     (MISRAC2012-RULE_2_1-a) PASSED
```

```
There shall be no unreachable code after 'return', 'break', 'continue', and
     'goto' statements (MISRAC2012-RULE_2_1-b) PASSED
There shall be no unreachable code in "if/else/while/for" block
     (MISRAC2012-RULE_2_1-c) PASSED
There shall be no unreachable code in switch statement
     (MISRAC2012-RULE_2_1-d) PASSED
There shall be no unreachable code in 'for' loop (MISRAC2012-RULE_2_1-e)
     PASSED
There shall be no unreachable code after 'if' or 'switch' statement
     (MISRAC2012-RULE_2_1-f) PASSED
There shall be no unreachable code after "if" or
     "switch" statement inside while/for/do...while loop
     (MISRAC2012-RULE_2_1-g) PASSED
All non-null statements shall either have at least one side-effect however
     executed or cause control flow to change (MISRAC2012-RULE_2_2-a) PASSED
Avoid unused values (MISRAC2012-RULE_2_2-b) FAILED
A function should not contain unused type declarations
     (MISRAC2012-RULE_2_3-a) PASSED
A source file should not contain unused type declarations
     (MISRAC2012-RULE_2_3-b) PASSED
A function should not contain unused local tag declarations
     (MISRAC2012-RULE_2_4-a) PASSED
A source file should not contain unused tag declarations
     (MISRAC2012-RULE_2_4-b) PASSED
A source file should not contain unused macro definitions
     (MISRAC2012-RULE_2_5-a) PASSED
A function should not contain unused label declarations
     (MISRAC2012-RULE_2_6-a) PASSED
There should be no unused parameters in functions (MISRAC2012-RULE_2_7-a)
     PASSED
The character sequence /0-DISABLE_NOUVEAU.TXT /apps /backup-ldapmatch /bin
     /boot /cds /cgroup /depot /dev /eng /etc /file /grid /hls /home /icd
     /lan /lib /lib64 /local /lost+found /media /misc /mnt /net /newdepot
     /opt /proc /process /project /projects /root /rscratch /sbin /scratch
     /selinux /servers /srv /sys /test /tmp /userspace /usr /var /vols shall
     not be used within a C-style comment (MISRAC2012-RULE_3_1-a) PASSED
The character sequence // shall not be used within a C-style comment
     (MISRAC2012-RULE_3_1-b) PASSED
The character sequence /0-DISABLE_NOUVEAU.TXT /apps /backup-ldapmatch /bin
     /boot /cds /cgroup /depot /dev /eng /etc /file /grid /hls /home /icd
     /lan /lib /lib64 /local /lost+found /media /misc /mnt /net /newdepot
     /opt /proc /process /project /projects /root /rscratch /sbin /scratch
     /selinux /servers /srv /sys /test /tmp /userspace /usr /var /vols shall
     not be used within a C++-style comment (MISRAC2012-RULE_3_1-c) PASSED
Line-splicing shall not be used in // comments (MISRAC2012-RULE_3_2-a)
     PASSED
Octal and hexadecimal escape sequences shall be terminated
     (MISRAC2012-RULE_4_1-a) PASSED
Trigraphs shall not be used (MISRAC2012-RULE_4_2-a) PASSED
External identifiers shall be distinct (MISRAC2012-RULE_5_1-a) PASSED
Identifiers declared in the file scope and in the same name space shall be
     distinct (c90) (MISRAC2012-RULE_5_2-a) PASSED
Identifiers declared in the file scope and in the same name space shall be
     distinct (c99) (MISRAC2012-RULE_5_2-b) PASSED
Identifiers declared in the same block scope and name space shall be
     distinct (c90) (MISRAC2012-RULE_5_2-c) PASSED
Identifiers declared in the same block scope and name space shall be
     distinct (c99) (MISRAC2012-RULE_5_2-d) PASSED
Identifier declared in a local or function prototype scope shall not hide an
     identifier declared in a global or namespace scope
     (MISRAC2012-RULE_5_3-a) PASSED
Identifiers declared in an inner local scope should not hide identifiers
     declared in an outer local scope (MISRAC2012-RULE_5_3-b) PASSED
The name of a macro should be distinct from the names of its parameters(c90)
```

```
     (MISRAC2012-RULE_5_4-a) PASSED
The name of a macro should be distinct from the names of its parameters(c99)
     (MISRAC2012-RULE_5_4-b) PASSED
The name of a macro should be distinct from the names of other macros that
     are currently defined(c90) (MISRAC2012-RULE_5_4-c) FAILED
The name of a macro should be distinct from the names of other macros that
     are currently defined(c99) (MISRAC2012-RULE_5_4-d) FAILED
The names of macros that exist prior to preprocessing should be distinct
     from the identifiers that exist after preprocessing (c90)
     (MISRAC2012-RULE_5_5-a) PASSED
The names of macros that exist prior to preprocessing should be distinct
     from the identifiers that exist after preprocessing (c99)
     (MISRAC2012-RULE_5_5-b) PASSED
Do not reuse typedef names (MISRAC2012-RULE_5_6-a) PASSED
Do not reuse typedef names as a typedef name (MISRAC2012-RULE_5_6-b) PASSED
A tag name shall not be reused for other purpose within the program
     (MISRAC2012-RULE_5_7-a) PASSED
A tag name shall not be reused to define a different tag
     (MISRAC2012-RULE_5_7-b) PASSED
Identifiers that define objects or functions with external linkage shall be
     unique (MISRAC2012-RULE_5_8-a) PASSED
No object or function identifier with static storage duration should be
     reused (MISRAC2012-RULE_5_9-a) PASSED
No object or function identifier with static storage duration should be
     reused (MISRAC2012-RULE_5_9-b) PASSED
Bit fields shall only be defined to be of type unsigned int or signed int
     (MISRAC2012-RULE_6_1-a) PASSED
Named bit-fields with signed integer type shall have a length of more than
     one bit (MISRAC2012-RULE_6_2-a) PASSED
Octal constants (other than zero) shall not be used (MISRAC2012-RULE_7_1-a)
     PASSED
A 'U' suffix shall be applied to all constants of unsigned type
     (MISRAC2012-RULE_7_2-a) PASSED
Use capital 'L' instead of lowercase 'l' to indicate long
     (MISRAC2012-RULE_7_3-a) PASSED
A string literal shall not be modified (MISRAC2012-RULE_7_4-a) PASSED
Whenever a function is declared or defined, its type shall be explicitly
     stated (MISRAC2012-RULE_8_1-a) PASSED
Whenever an object is declared or defined, its type shall be explicitly
     stated (MISRAC2012-RULE_8_1-b) PASSED
An inline function shall be declared with the static storage class
     (MISRAC2012-RULE_8_10-a) PASSED
When an array is declared with external linkage, its size shall be stated
     explicitly or defined implicitly by initialisation
     (MISRAC2012-RULE_8_11-a) PASSED
Within an enumerator list, the value of an implicitly-specified enumeration
     constant shall be unique (MISRAC2012-RULE_8_12-a) PASSED
A pointer parameter in a function prototype should be declared as pointer to
     const if the pointer is not used to modify the addressed object
     (MISRAC2012-RULE_8_13-a) FAILED
Declare a type of parameter as typedef to pointer to const if the pointer is
     not used to modify the addressed object (MISRAC2012-RULE_8_13-b) PASSED
The restrict type qualifier shall not be used (MISRAC2012-RULE_8_14-a)
     PASSED
Identifiers shall be given for all of the parameters in a function prototype
     declaration (MISRAC2012-RULE_8_2-a) PASSED
Function types shall have named parameters (MISRAC2012-RULE_8_2-b) PASSED
Function types shall be in prototype form (MISRAC2012-RULE_8_2-c) PASSED
If objects or functions are declared more than once their types shall be
     compatible (MISRAC2012-RULE_8_3-a) PASSED
The identifiers used in the declaration and definition of a function shall
     be identical (MISRAC2012-RULE_8_3-b) FAILED
All declarations of an object or function shall have compatible types
     (MISRAC2012-RULE_8_3-c) PASSED
```

```
A declaration shall be visible when an object or function with external
     linkage is defined (MISRAC2012-RULE_8_4-a) PASSED
If objects or functions are declared more than once their types shall be
     compatible (MISRAC2012-RULE_8_4-b) PASSED
An external object or function shall not have more than one non-defining
     declaration in translation unit (MISRAC2012-RULE_8_5-a) PASSED
An identifier with external linkage shall have exactly one external
     definition (MISRAC2012-RULE_8_6-a) WAIVED
Functions and objects should not be defined with external linkage if they
     are referenced in only one translation unit (MISRAC2012-RULE_8_7-a)
     FAILED
The static storage class specifier shall be used in definitions and
     declarations of objects and functions that have internal linkage
     (MISRAC2012-RULE_8_8-a) PASSED
Objects shall be defined at block scope if they are only accessed from
     within a single function (MISRAC2012-RULE_8_9-a) PASSED
Avoid use before initialization (MISRAC2012-RULE_9_1-a) PASSED
The initializer for an aggregate or union shall be enclosed in braces
     (MISRAC2012-RULE_9_2-a) PASSED
Arrays shall not be partially initialized (MISRAC2012-RULE_9_3-a) PASSED
An element of an object shall not be initialized more than once
     (MISRAC2012-RULE_9_4-a) PASSED
Where designated initializers are used to initialize an array object the
     size of the array shall be specified explicitly (MISRAC2012-RULE_9_5-a)
     PASSED
```

5.2.6. Parasoft DTP Engine for C/C++ Analysis - HIS Source Code Metrics

```
A function shall have at most one exit point (CODSTA-91) PASSED
Avoid functions with more than 5 parameters (METRICS-15) PASSED
Follow the Cyclomatic Complexity limit of 10 (METRICS-18) FAILED
A global function should not be called from more than 5 different functions
     (METRICS-36) WAIVED
A function should not call more than 7 different functions (METRICS-37)
     FAILED
The number of statements within function should be in range 1 - 50
     (METRICS-38) PASSED
The value of VOCF metric for a function should not be higher than 4
     (METRICS-39) FAILED
Statements within function should not be nested deeper than 4 levels
     (METRICS-40) FAILED
The number of blocks of comments before and inside function to the number of
     statements in function should be > 0.2 (METRICS-41) WAIVED
The goto statement shall not be used (MISRA2004-14_4) PASSED
Functions shall not call themselves, either directly or indirectly
     (MISRA2004-16_2) PASSED
```

Chapter 6. Test Code Static Analysis

6.1. Static Analysis summary

Static Analysis for test code runs Parasoft's recommended rules against code used for driver verification.
Summary:

Recommended Rules: 55 rules, 54 passed, 1 waived, 0 failed

6.2. Static Analysis

All Static Analysis was performed using DTP Engine for C/C++ 10.3.4 by Parasoft. This analisys was conducted in a 64-bit environment. No rules were disabled. All violations are marked as FAILED in this report.

6.2.1. Parasoft DTP Engine for C/C++ Analysis - Recommended Rules tests code

```
Do not pass negative values to functions expecting non-negative arguments
     (BD-API-NEGPARAM) PASSED
Always catch exceptions (BD-PB-EXCEPT) PASSED
Avoid use before initialization (BD-PB-NOTINIT) PASSED
Avoid null pointer dereferencing (BD-PB-NP) PASSED
Avoid buffer overflow due to defining incorrect format limits
     (BD-PB-OVERFFMT) PASSED
Avoid overflow due to reading a not zero terminated string (BD-PB-OVERFNZT)
     PASSED
Avoid overflow when reading from a buffer (BD-PB-OVERFRD) PASSED
Avoid overflow when writing to a buffer (BD-PB-OVERFWR) PASSED
Avoid division by zero (BD-PB-ZERO) PASSED
Avoid accessing arrays out of bounds (BD-PB-ARRAY) PASSED
Avoid conditions that always evaluate to the same value (BD-PB-CC) PASSED
Do not check for null after dereferencing (BD-PB-DEREF) PASSED
Suspicious setting of stream flags (BD-PB-STREAMFLAGS) PASSED
Restore stream format (BD-PB-STREAMFMT) PASSED
Properly deallocate dynamically allocated resources (BD-RES-BADDEALLOC)
     PASSED
Do not use resources that have been freed (BD-RES-FREE) PASSED
Do not free resources using invalid pointers (BD-RES-INVFREE) PASSED
Ensure resources are freed (BD-RES-LEAKS) PASSED
Avoid double locking (BD-TRS-DLOCK) PASSED
Avoid race conditions when using fork and file descriptors (BD-TRS-FORKFILE)
     PASSED
Do not abandon unreleased locks (BD-TRS-LOCK) PASSED
Do not acquire locks in different order (BD-TRS-ORDER) PASSED
Avoid race conditions while checking for the existence of a symbolic link
     (BD-TRS-SYMLINK) PASSED
Do not use blocking functions while holding a lock (BD-TRS-TSHL) PASSED
Avoid function duplication (CDD-DUPM) WAIVED
Local variables should not use the same names as member variables
     (CODSTA-44) PASSED
Pointer should not be compared with NULL using relational operators <, >,
     >=, <= (CODSTA-147) PASSED
Do not use string literals as operands of equality or relational operators
     (CODSTA-148) PASSED
Avoid infinite loops (CODSTA-82) PASSED
Constructors allowing for conversion should be made explicit (CODSTA-CPP-04)
     PASSED
Throw by value, catch by reference (EXCEPT-02) PASSED
Do not throw from within destructor (EXCEPT-03) PASSED
```

```
All member variables should be initialized in constructor (INIT-06) PASSED
McCabe Cyclomatic Complexity (METRIC.CC) PASSED
Nested Blocks Depth (METRIC.NBD) PASSED
Floating-point expressions shall not be tested for equality or inequality
     (MISRA2004-13_3) PASSED
All exit paths from a function with non-void return type shall have an
     explicit return statement with an expression (MISRA2004-16_8) PASSED
The address of an object with automatic storage shall not be returned from a
     function (MISRA2004-17_6_a) PASSED
Do not invoke malloc/realloc for objects having constructors (MRM-08) PASSED
Declare a copy assignment operator for classes with dynamically allocated
     memory (MRM-37) PASSED
Declare a copy constructor for classes with dynamically allocated memory
     (MRM-38) PASSED
Never provide brackets ([]) for delete when deallocating non-arrays (MRM-35)
     PASSED
Always provide empty brackets ([]) for delete when deallocating arrays
     (MRM-36) PASSED
Do not use 'delete' on pointers to a void type (MRM-51) PASSED
Class cannot inherit other class more than once unless it is virtual
     inheritance (OOP-03) PASSED
Avoid calling virtual functions from constructors (OOP-16) PASSED
If a class has virtual functions it shall have a virtual destructor (OOP-23)
     PASSED
Pass objects by reference instead of by value (OPT-14) PASSED
Do not call delete on non-pointers (PB-13) PASSED
Properly terminate character strings (PB-21) PASSED
Do not cast from or to incomplete class at the point of casting (PB-54)
     PASSED
Do not delete objects with incomplete class at the point of deletion (PB-55)
     PASSED
Boolean condition always evaluates to the same value due to enumeration with
     only zero or only non-zero constants (PB-68) PASSED
Suspicious argument to malloc (PB-60) PASSED
Pointer arithmetic performed on freshly allocated memory (PB-61) PASSED
```