cadence

Cadence Driver Verification Plan and Test Report

Product Version 5.7.1

April 2020

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:82f725edc9078f92a4ed7cea3ba0961e

Table of Contents

List of Figures

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.

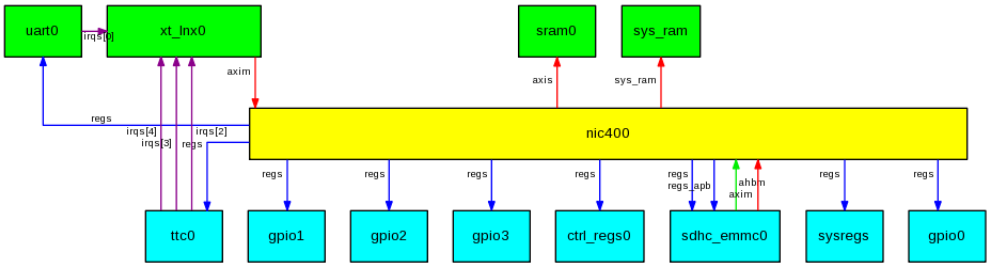
Negative For this type of tests, the required operation will not be performed, and driver shall return an error.

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.



Chapter 3. Test Results

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

3.1. Functional tests in RTL environment

3.1.1. 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".

Functional tests - eMMC: 20 calls, 18 passed, 2 not supported

Functional tests: 2 calls, 1 passed, 1 not supported

3.1.2. Test Details

3.1.2.1. Functional tests - eMMC

```
Boot test PASSED
PhyTrainingMmc PASSED
SimpleTest PASSED
SubcommandTest NOT SUPPORTED
InfiniteReadTest PASSED
InfiniteWriteTest PASSED
SingleSectorTest PASSED
DMATest PASSED
ADMA3Test NOT SUPPORTED
LowClockFreqTest PASSED
HSTest PASSED
BusWidthTest PASSED
MmchighSpeedTest PASSED
CQ Direct command test PASSED
CQ Data transfer test PASSED
CQ Direct command pooling mode test PASSED
CQ Data transfer pooling mode test PASSED
CQ Split data across queues test PASSED
CQ Split data across descriptors test PASSED
CQ Interrupt coalescing test PASSED
```

3.1.2.2. Functional tests

```
ErrorInjectionTest PASSED
subcommandErrorInjectionTest NOT SUPPORTED
```

3.2. Mechanical tests

3.2.1. 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".

nullptr: 94 calls, 94 passed

ranges: 94 calls, 56 passed, 38 not supported

3.2.2. Test Details

3.2.2.1. nullptr

```
probe PASSED
init PASSED
start PASSED
stop PASSED
execCardCommand PASSED
deviceDetach PASSED
deviceAttach PASSED
abort PASSED
standBy PASSED
configure PASSED
isr PASSED
configureHighSpeed PASSED
checkSlots PASSED
checkInterrupt PASSED
configureAccessMode PASSED
tuning PASSED
clockGeneratorSelect PASSED
presetValueSwitch PASSED
configureDriverStrength PASSED
memoryCardLoadDriver PASSED
memoryCardDataTransfer PASSED
memoryCardDataTransfer2 PASSED
memoryCardConfigure PASSED
memoryCardDataErase PASSED
memCardPartialDataXfer PASSED
memCardInfXferStart PASSED
memCardInfXferContinue PASSED
memCardInfXferFinish PASSED
memCardDataXferNonBlock PASSED
memCardFinishXferNonBlock PASSED
phySettingsSd3 PASSED
phySettingsSd4 PASSED
writePhySet PASSED
readPhySet PASSED
readCardStatus PASSED
selectCard PASSED
resetCard PASSED
execCmd55Command PASSED
accessCccr PASSED
readCsd PASSED
readExCsd PASSED
getTupleFromCis PASSED
readSdStatus PASSED
setDriverStrength PASSED
execSetCurrentLimit PASSED
mmcSwitch PASSED
mmcSetExtCsd PASSED
mmcSetBootPartition PASSED
mmcSetPartAccess PASSED
mmcSetBootAck PASSED
mmcExecuteBoot PASSED
mmcGetParitionBootSize PASSED
getInterfaceType PASSED
getDeviceState PASSED
memoryCardGetSecCount PASSED
setDriverData PASSED
getDriverData PASSED
simpleInit PASSED
resetHost PASSED
```

```
getRca PASSED
cQEnable PASSED
cQDisable PASSED
cQGetInitConfig PASSED
cQGetUnusedTaskId PASSED
cQStartExecuteTask PASSED
cQAttachRequest PASSED
cQExecuteDcmdRequest PASSED
cQGetDirectCmdConfig PASSED
cQSetDirectCmdConfig PASSED
cQSetIntCoalescingConfig PASSED
cQGetIntCoalescingConfig PASSED
cQGetIntCoalescingTimeoutBase PASSED
cQStartExecuteTasks PASSED
cQHalt PASSED
cQTaskDiscard PASSED
cQAllTasksDiscard PASSED
cQResetIntCoalCounters PASSED
cQSetResponseErrorMask PASSED
cQGetResponseErrorMask PASSED
getBaseClk PASSED
waitForRequest PASSED
setCPhyConfigIoDelay PASSED
getCPhyConfigIoDelay PASSED
setCPhyConfigLvsi PASSED
getCPhyConfigLvsi PASSED
setCPhyConfigDfiRd PASSED
getCPhyConfigDfiRd PASSED
setCPhyConfigOutputDelay PASSED
getCPhyConfigOutputDelay PASSED
cPhyDllReset PASSED
setCPhyExtMode PASSED
getCPhyExtMode PASSED
setCPhySdclkAdj PASSED
getCPhySdclkAdj PASSED
```

3.2.2.2. ranges

```
probe PASSED
init NOT SUPPORTED
start PASSED
stop PASSED
execCardCommand NOT SUPPORTED
deviceDetach PASSED
deviceAttach PASSED
abort NOT SUPPORTED
standBy NOT SUPPORTED
configure PASSED
isr PASSED
configureHighSpeed PASSED
checkSlots PASSED
checkInterrupt NOT SUPPORTED
configureAccessMode PASSED
tuning PASSED
clockGeneratorSelect PASSED
presetValueSwitch PASSED
configureDriverStrength PASSED
memoryCardLoadDriver PASSED
memoryCardDataTransfer PASSED
memoryCardDataTransfer2 NOT SUPPORTED
memoryCardConfigure NOT SUPPORTED
memoryCardDataErase NOT SUPPORTED
memCardPartialDataXfer NOT SUPPORTED
```

```
memCardInfXferStart NOT SUPPORTED
memCardInfXferContinue NOT SUPPORTED
memCardInfXferFinish NOT SUPPORTED
memCardDataXferNonBlock NOT SUPPORTED
memCardFinishXferNonBlock NOT SUPPORTED
phySettingsSd3 NOT SUPPORTED
phySettingsSd4 NOT SUPPORTED
writePhySet NOT SUPPORTED
readPhySet NOT SUPPORTED
readCardStatus NOT SUPPORTED
selectCard NOT SUPPORTED
resetCard NOT SUPPORTED
execCmd55Command NOT SUPPORTED
accessCccr PASSED
readCsd NOT SUPPORTED
readExCsd NOT SUPPORTED
getTupleFromCis PASSED
readSdStatus NOT SUPPORTED
setDriverStrength PASSED
execSetCurrentLimit PASSED
mmcSwitch NOT SUPPORTED
mmcSetExtCsd NOT SUPPORTED
mmcSetBootPartition NOT SUPPORTED
mmcSetPartAccess NOT SUPPORTED
mmcSetBootAck PASSED
mmcExecuteBoot NOT SUPPORTED
mmcGetParitionBootSize NOT SUPPORTED
getInterfaceType PASSED
getDeviceState PASSED
memoryCardGetSecCount NOT SUPPORTED
setDriverData PASSED
getDriverData PASSED
simpleInit NOT SUPPORTED
resetHost PASSED
getRca NOT SUPPORTED
cQEnable NOT SUPPORTED
cQDisable PASSED
cQGetInitConfig PASSED
cQGetUnusedTaskId PASSED
cQStartExecuteTask PASSED
cQAttachRequest NOT SUPPORTED
cQExecuteDcmdRequest NOT SUPPORTED
cQGetDirectCmdConfig PASSED
cQSetDirectCmdConfig PASSED
cQSetIntCoalescingConfig PASSED
cQGetIntCoalescingConfig PASSED
cQGetIntCoalescingTimeoutBase PASSED
cQStartExecuteTasks PASSED
cQHalt PASSED
cQTaskDiscard PASSED
cQAllTasksDiscard PASSED
cQResetIntCoalCounters PASSED
cQSetResponseErrorMask PASSED
cQGetResponseErrorMask PASSED
getBaseClk PASSED
waitForRequest NOT SUPPORTED
setCPhyConfigIoDelay PASSED
getCPhyConfigIoDelay PASSED
setCPhyConfigLvsi PASSED
getCPhyConfigLvsi PASSED
setCPhyConfigDfiRd PASSED
getCPhyConfigDfiRd PASSED
setCPhyConfigOutputDelay PASSED
getCPhyConfigOutputDelay PASSED
```

```
cPhyDllReset PASSED
setCPhyExtMode PASSED
getCPhyExtMode PASSED
setCPhySdclkAdj PASSED
getCPhySdclkAdj PASSED
```

Chapter 4. Static Analysis

4.1. Static Analysis summary

Static Analysis consists of a check of Parasoft's recommended rules:

Recommended Rules: 53 rules, 53 passed

4.2. 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. For static analysis the following sets of rules were defined:

- Parasoft recommended rules

4.2.1. 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
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
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
```