



SD Host Firmware Driver Quick Start Guide

Product Version 5.7.1

April 2020

© 1996-2020 Cadence Design Systems, Inc. All rights reserved.

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

Trademarks: Trademarks and service marks of Cadence Design Systems, Inc. contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 800.862.4522. All other trademarks are the property of their respective holders.

Restricted Permission: This document is protected by copyright law and international treaties and contains trade secrets and proprietary information owned by Cadence. Unauthorized reproduction or distribution of this document, or any portion of it, may result in civil and criminal penalties. Except as specified in this permission statement, this document may not be copied, reproduced, modified, published, uploaded, posted, transmitted, or distributed in any way, without prior written permission from Cadence. This document contains the proprietary and confidential information of Cadence or its licensors, and is supplied subject to, and may be used only in accordance with, a written agreement between Cadence and its customer.

Unless otherwise agreed to by Cadence in writing, this statement grants Cadence customers permission to print one (1) hard copy of this document subject to the following conditions:

1. This document may not be modified in any way.
2. Any authorized copy of this document or portion thereof must include all original copyright, trademark, and other proprietary notices and this permission statement.
3. The information contained in this document cannot be used in the development of like products or software, whether for internal or external use, and shall not be used for the benefit of any other party, whether or not for consideration.

Disclaimer: INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND DOES NOT REPRESENT A COMMITMENT ON THE PART OF CADENCE. EXCEPT AS MAY BE EXPLICITLY SET FORTH IN A WRITTEN AGREEMENT BETWEEN CADENCE AND ITS CUSTOMER, CADENCE DOES NOT MAKE, AND EXPRESSLY DISCLAIMS, ANY REPRESENTATIONS OR WARRANTIES AS TO THE COMPLETENESS, ACCURACY OR USEFULNESS OF THE INFORMATION CONTAINED IN THIS DOCUMENT. CADENCE DOES NOT WARRANT THAT USE OF SUCH INFORMATION WILL NOT INFRINGE ANY THIRD PARTY RIGHTS, AND CADENCE DISCLAIMS ALL IMPLIED WARRANTIES, INCLUDING MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. CADENCE DOES NOT ASSUME ANY LIABILITY FOR DAMAGES OR COSTS OF ANY KIND THAT MAY RESULT FROM USE OF SUCH INFORMATION. CADENCE CUSTOMER HAS COMPLETE CONTROL AND FINAL DECISION-MAKING AUTHORITY OVER ALL ASPECTS OF THE DEVELOPMENT, MANUFACTURE, SALE AND USE OF CUSTOMER'S PRODUCT, INCLUDING, BUT NOT LIMITED TO, ALL DECISIONS WITH REGARD TO DESIGN, PRODUCTION, TESTING, ASSEMBLY, QUALIFICATION, CERTIFICATION, INTEGRATION OF CADENCE PRODUCTS, INSTRUCTIONS FOR USE, LABELING AND DISTRIBUTION, AND CADENCE EXPRESSLY DISAVOWS ANY RESPONSIBILITY WITH REGARD TO ANY SUCH DECISIONS REGARDING CUSTOMER'S PRODUCT.

Restricted Rights: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227- 14 and DFAR252.227-7013 et seq. or its successor.

1. Overview

This document describes the basics of how to get up and running with the Cadence IP Core Driver.

- Prepare the Cadence Platform Services (CPS)
- Prepare interrupt handling
- Build the core-driver
- Use the core-driver to initialize the Cadence IP

2. Details

Here's how to get started:

1. *Prepare the Cadence Platform Services:*

This is covered in detail in the Cadence Driver Porting Guide, which is available at [doc/porting/porting_guide.pdf](#). A sample implementation for bare-metal systems is available in `RefCode/cps_bm.c`. You may be able to use these without modification.

CPS is a platform-specific code to connect a given Cadence driver to the rest of your system. Not all of the features of CPS are required for this driver. You only need to implement the functions for uncached access to memory:

```
CPS_UncachedRead8, CPS_UncachedRead16, CPS_UncachedRead32,  
CPS_UncachedWrite8, CPS_UncachedWrite16 and CPS_UncachedWrite32
```

These functions will be used by the core driver to access the memory in a portable manner. `CPS_CacheFlush()` / `CPS_CacheInvalidate()` are not used by this driver. However, this means that you must ensure cache consistency of any areas of memory passed to the driver for use in DMA transfers.

After this step you should have an object file implementing the CPS for your target system which you can link with later.

2. *Prepare interrupt handling*

Driver can work in two modes interrupt mode and polling mode.

To work in interrupt mode user needs to call *start* function. In this mode interrupts are used to handle controller requests. If function is not called then driver works in polling mode.

3. Build the core-driver

All driver sources are provided in the CoreDriver directory. You may need to modify `environemnt_config.h` file to ajust driver to your platform.

4. Use the core-driver to initialize the Cadence IP

There are two samples available in RefCode folder. Each of them executes the same initiaization steps.

The sample code is designed to require only minor changes before it can be used. You will however need to change some default values to match your hardware. Change these

```
/* This should be the base address of
   the Cadence IP core on your system */
#define SD_REG_BASE          0x50000000
```

The sample code then calls *probe* to confirm that the hardware is available at the specified address, and to determine the size of memory which is required for private data and dma descriptors. This memory must be provided by the code calling the driver, and may be allocated at compile time (e.g. by an array allocation as shown in the sample code) or dynamically. The private data and addresses(virtual and logical) will be passed to the driver as a parameter for each call to the API.

The sample code then calls *init* to initialize the driver.

As well as initializing the driver, the sample code also displays some diagnostic information using `stdio`. If `stdio` is not available on your system, these calls should be stubbed out or replaced with an alternative (e.g. write to memory or local storage.)

After modifying the sample code to suit your environment, you should compile for your target system and link with the driver and CPS objects prepared previously, then download and run the code on your target system. This will be hardware and operating system specific, and is not covered in this documentation.