



ALPHA DATA

**ADM-PCIE-9H3 Support &
Development Kit
Release: 0.1.1**

**Document Revision: 1.1
24th January 2019**

**© 2019 Copyright Alpha Data Parallel Systems Ltd.
All rights reserved.**

This publication is protected by Copyright Law, with all rights reserved. No part of this publication may be reproduced, in any shape or form, without prior written consent from Alpha Data Parallel Systems Ltd.

Head Office

Address: 4 West Silvermills Lane
Edinburgh, EH3 5BD, UK
Telephone: +44 131 558 2600
Fax: +44 131 558 2700
email: sales@alpha-data.com
website: <http://www.alpha-data.com>

US Office

611 Corporate Circle, Suite H
Golden, CO 80401
(303) 954 8768
(866) 820 9956 - toll free
sales@alpha-data.com
<http://www.alpha-data.com>

All trademarks are the property of their respective owners.

Table Of Contents

1	Introduction	1
1.1	<i>Structure of this package</i>	2
1.2	<i>The ADXDMA Driver</i>	2
2	Recommended Vivado versions	3
3	Development operating system support	4
3.1	<i>Windows</i>	4
3.2	<i>Linux</i>	4
4	Conventions used in this SDK	5
4.1	<i>Version numbers</i>	5
4.2	<i>Common HDL code</i>	5
4.3	<i>Vivado IP repositories</i>	5
4.4	<i>Example FPGA designs</i>	5
5	Associated documents	8
6	Release history	9
6.1	<i>Release 0.1.0</i>	9
6.2	<i>Release 0.1.1</i>	9

List of Tables

Table 1	Recommended Vivado version by example FPGA design and silicon revision	3
Table 2	Vivado Tcl scripts used in example FPGA designs	6

List of Figures

Figure 1	Structure of the SDK	2
Figure 2	Structure of an example FPGA design	6

1 Introduction

The **ADM-PCIE-9H3 Support & Development Kit (SDK)** is a set of resources for FPGA designers and software engineers working with Alpha Data's ADM-PCIE-9H3 reconfigurable computing card. The latest release of the ADM-PCIE-9H3 SDK can be found at:

https://support.alpha-data.com/Downloads.aspx?fldr_req=Downloads/admpcie9h3/sdk

Note

This release of the ADM-PCIE-9H3 SDK supports engineering silicon (ES1) only. At the time of writing, production silicon for the Xilinx VU33P and VU35P FPGAs is not generally available.

The resources of the ADM-PCIE-9H3 SDK include:

- Resources for developing application software for a machine that hosts Alpha Data reconfigurable computing hardware, using the [ADXDMA Driver](#):
 - C/C++ header files and libraries which provide Application Programming Interfaces (APIs).
 - Documentation about Application Programming Interfaces (APIs).
 - Utilities (with source code), which make use of the Application Programming Interfaces (APIs).
- Example FPGA designs and host programs (with source code) demonstrating various features of the ADM-PCIE-9H3:
 - Standalone HBM Test FPGA Design**
Demonstrates how to use Xilinx's Ultrascale+ HBM IP with the ADM-PCIE-9H3.
 - DMA Demonstration FPGA Design**
Demonstrates high-performance DMA using the Xilinx XDMA (PCI Express) IP together with Alpha Data's [ADXDMA Driver](#).
 - Host Interface to HBM FPGA Design**
Demonstrates combining the Xilinx XDMA (PCI Express) IP with the Xilinx Ultrascale+ HBM IP in order to create a host interface that permits access to the on-chip HBM from the host system.
A host program that uses the [ADXDMA Driver](#) demonstrates DMA data transfer between host memory and the HBM.
 - Host Interface to SPI Flash Design**
Demonstrates combining the Xilinx XDMA (PCI Express) IP with the Xilinx AXI SPI IP in order to create a host interface that permits the host CPU to access the on-board SPI Flash that is used to configure the FPGA.
The [ADXDMA Driver](#) includes a utility `adxdma_spi` that allows the host CPU to erase, program and verify SPI Flash chips via the Xilinx AXI SPI IP.
 - IBERT FPGA Design**
Makes use of the Xilinx IBERT IP to provide ready-to-use IBERT bitstreams that can be used to test the QSFP-DD and OpenCAPI connectors of the ADM-PCIE-9H3.
- IP and common HDL code provided by Alpha Data:
 - Board Control Interface IP (ADM-PCIE-9H3-BCI)**
Provides an AXI4 Lite interface to the board's microcontroller, among other functions.
 - Common HDL code (i.e. not specific to the ADM-PCIE-9H3), used by the example FPGA designs.

1.1 Structure of this package

The directories making up the ADM-PCIE-9H3 SDK are organised as in [Figure 1](#) below:

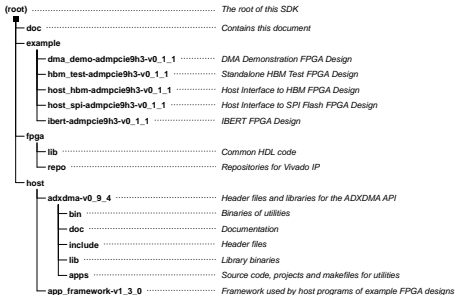


Figure 1 : Structure of the SDK

1.2 The ADXDMA Driver

The **ADXDMA Driver** is a kernel-mode driver for the PCI Express Endpoint in the Xilinx XDMA IP, which provides a user-mode application programming interface (API). The latest release can be found here:

https://support.alpha-data.com/Downloads.aspx?fldr_req=Downloads/adxdma

2 Recommended Vivado versions

For this release of the ADM-PCIE-9H3 SDK, the Xilinx Vivado toolset version 2018.1 to 2018.3 (inclusive) can be used for building and simulating the example FPGA designs for the ES1 silicon revision. Recommended Vivado versions for the example FPGA designs are as follows:

Example FPGA design	Short name	Silicon revision	Recommended Vivado version
DMA Demonstration FPGA Design	dma_demo	ES1	2018.3
Standalone HBM Test FPGA Design	hbm_test	ES1	2018.3
Host Interface to HBM FPGA Design	host_hbm	ES1	2018.3
Host Interface to SPI Flash FPGA Design	host_spi	ES1	2018.3
IBERT FPGA Design	ibert	ES1	2018.3

Table 1 : Recommended Vivado version by example FPGA design and silicon revision

Please refer to the documentation for each example FPGA design for further details of any known issues that affect them.

3 Development operating system support

3.1 Windows

Generally speaking, Alpha Data's Windows software, when supplied in binary form, is compatible with Windows XP and later. However, the choice of Windows operating system used for development mainly depends upon which releases of Microsoft Visual Studio and/or Xilinx Vivado are chosen for a project.

When developing an FPGA design, the Xilinx Vivado toolset is used and therefore a Windows operating system must be capable of running Vivado. As of writing, Vivado 2018.3 is the current release, and as per the [Vivado 2018.3 release notes](#), Windows 7 SP1 and Windows 10 are recommended.

Vivado path length issue

In Windows, Vivado requires that path lengths of files are no greater than the **MAX_PATH** Win32 constant, which is 260 characters (including the NUL character used to terminate a string). This limit is easily exceeded when a Vivado project uses IP (cores) and the path length of the Vivado project file (.xpr) exceeds about 80 characters. Exceeding the **MAX_PATH** limit can result in otherwise inexplicable failures when implementing an FPGA design in Vivado.

The recommended workaround for this issue is to use the **subst** command to map a drive letter (e.g. **Z:**) to the root of this SDK. If done correctly, the result is the existence of directories **Z:\doc**, **Z:\example**, **Z:\fpga** etc.

When developing software to run on a host machine, Microsoft Visual Studio is likely to be used for building applications. Therefore, the Windows operating system must be capable of running a particular version of Microsoft Visual Studio. For Microsoft Visual Studio 2017, Windows 7, Windows 8.1 and Windows 10 are recommended.

3.2 Linux

Alpha Data generally does not supply binaries for Linux because of the large number of architectures and configurations that exist across various Linux distributions. Source code is provided, however, and can be built for most Linux distributions.

When developing an FPGA design, the Xilinx Vivado toolset is used, and therefore the supported Linux distributions depend upon the version of Vivado chosen for a project. As of writing, Vivado 2018.3 is the latest version, and as per the [Vivado 2018.3 release notes](#), the following Linux distributions are recommended:

- Red Hat Enterprise Workstation / Server 7.2, 7.3, 7.4 & 7.5 (64-bit)
- Red Hat Enterprise Workstation 6.6, 6.7, 6.8 & 6.9 (64-bit)
- SUSE Linux Enterprise 11.4 & 12.3 (64-bit)
- CentOS 6.7, 6.8, 6.9, 7.2, 7.3, 7.4 & 7.5 (64-bit)
- Ubuntu Linux 16.04.4 LTS & 18.04 LTS (64-bit)

Linux distributions that are not listed might result in Vivado failing to work, or only partially working. They are not supported by Xilinx.

4 Conventions used in this SDK

4.1 Version numbers

If the name of a folder in the SDK has a suffix such as `v1_0_0`, it denotes a version number for that subtree as a whole.

4.2 Common HDL code

The folder `fpga/lib/` contains common HDL code used by certain example FPGA designs. Within this folder, each component (a collection of HDL source files) is a subtree whose name has a version suffix (for example, `v1_0_0`)

4.3 Vivado IP repositories

The folder `fpga/repo/` contains Vivado IP repositories used by certain example FPGA designs. Generally speaking, if this folder contains a subfolder `XXXX.Y`, it means that the IP-XACT definitions found within it are compatible with Vivado `XXXX.Y` or later.

The folder `fpga/repo/interfaces/` might exist in some SDKs. This folder contains IP-XACT interface definitions, which are generally compatible with any version of Vivado.

4.4 Example FPGA designs

The example FPGA designs, found in the `example` folder, consist of the following elements:

- Documentation describing the example FPGA design.
- FPGA-related:
 - HDL code (VHDL / Verilog / SystemVerilog).
 - Constraints (.xdc files).
 - Tcl scripts for generating Vivado projects.
Note that ready-made Vivado projects are **not** supplied; see below for an explanation of this.
 - Pre-built bitstreams, generally built with the latest version of Vivado available at time of publication.
- Host-related, if required for the example FPGA design in question:
 - C / C++ source and header files.
 - Project files for Microsoft Visual Studio and Makefiles for Linux & VxWorks, where supported for the example FPGA design in question.
 - Pre-built binaries, for Windows running on Intel architecture, if supported for the example FPGA design in question.

For Linux and VxWorks, there are so many possible architectures, binary formats, compilers, environmental issues and other build options that supplying binaries is deemed impractical. For these operating systems, the user must compile demonstration programs before running them.

As mentioned above, ready-made Vivado projects are **not** supplied in this SDK, but Tcl scripts for generating Vivado projects are provided. This is for several reasons, principally:

- Vivado projects using certain IP or certain features can be broken simply by being moved within the filesystem, and such a move would always occur when the user unpacks this SDK to the folder of his or her choice.
- Vivado projects using certain IP or certain features can be broken by being upgraded (to a newer version of Vivado). Vivado projects cannot in general be reliably downgraded (to an older version of Vivado).
- Tcl scripts for generating Vivado projects can determine the version of Vivado in use and apply

workarounds for known issues at time of publication.

The structure of each example FPGA design conforms to the following general template:

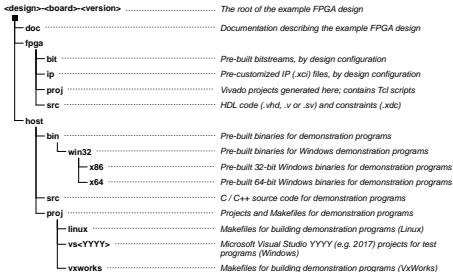


Figure 2 : Structure of an example FPGA design

Some folders may be omitted for certain example FPGA designs; for example, the **IBERT FPGA Design** does not have a **host** folder because it has no host interface and is driven solely by Vivado Hardware Manager.

Vivado Tcl script naming conventions

Within the **fpga/proj** folder of each example FPGA design, a number of Tcl scripts are provided. Most important are the Vivado project-generation scripts, whose filenames contain the **mkxpr** prefix and a suffix that identifies a particular *design configuration*. The commonly-provided Tcl scripts are given in [Table 2](#) below.

General form of filename	Purpose
build-<design configuration>.tcl	Implements the design in Vivado for a particular design configuration
debug-<design configuration>.tcl	Opens Vivado Hardware Manager and configures the FPGA in a card with the bitstream for a particular design configuration. These scripts can also be found in the fpga/bit/ folder of a given example FPGA design.
flash-<design configuration>.tcl	Opens Vivado Hardware Manager and writes the bitstream (a .mcs file) for a particular design configuration into the configuration Flash memory of an FPGA card.
mkxpr-<design configuration>.tcl	Generates a Vivado project for a particular design configuration.

Table 2 : Vivado Tcl scripts used in example FPGA designs (continued on next page)

General form of filename	Purpose
rebuild-<design configuration>.tcl	Re-implements the design in Vivado for a particular design configuration

Table 2 : Vivado Tcl scripts used in example FPGA designs

Design configurations allow for several variants to exist for a given example FPGA design. For example, in the IBERT FPGA Design, there are multiple design configurations, corresponding to different line rates and different FPGA dies (e.g. VU33P vs VU35P). In general, the total number of design configurations for a given example FPGA design is the result of enumerating the valid combinations of one or more "orthogonal" options.

5 Associated documents

- (1) ADXDMA API
(root)/host/adxdma-v0_9_4/doc/ad-ug-0105_v1_3.pdf
- (2) ADXDMA Board Control API
(root)/host/adxdma-v0_9_4/doc/ad-ug-0112_v1_2.pdf
- (3) ADXDMA Driver Utilities
(root)/host/adxdma-v0_9_4/doc/ad-ug-0110_v1_3.pdf
- (4) ADM-PCIE-9H3 DMA Demonstration FPGA Design
(root)/example/dma_demo-admpcie9h3-v0_1_1/doc/ad-ug-0122_v1_1.pdf
- (5) ADM-PCIE-9H3 Standalone HBM FPGA Design
(root)/example/hbm_test-admpcie9h3-v0_1_1/doc/ad-ug-0120_v1_1.pdf
- (6) ADM-PCIE-9H3 Host Interface to HBM FPGA Design
(root)/example/host_hbm-admpcie9h3-v0_1_1/doc/ad-ug-0124_v1_1.pdf
- (7) ADM-PCIE-9H3 Host Interface to SPI Flash FPGA Design
(root)/example/host_spi-admpcie9h3-v0_1_1/doc/ad-ug-0125_v1_1.pdf
- (8) ADM-PCIE-9H3 IBERT FPGA Design
(root)/example/ibert-admpcie9h3-v0_1_1/doc/ad-ug-0123_v1_1.pdf

6 Release history

6.1 Release 0.1.0

This is the first release of the ADM-PCIE-9H3 Support & Development Kit, with support for ES1 (only) VU33P and VU35P FPGAs.

6.2 Release 0.1.1

Corrects various issues encountered when attempting to generate projects for the example FPGA designs and build them using Vivado 2018.3:

- Fixed an issue in the **dma_demo** example FPGA design where, due new IP parameters introduced for the Xilinx Block Memory Generator IP in Vivado 2018.3, the read latency of the instantiated Ultra RAM is 1 (incorrect) instead of 5 (correct) and fails to meet timing requirements.
- Fixed an issue in the **hbm_test** and **host_hbm** example FPGA designs where, due the address map segment size changing from 512 MiB (Vivado 2018.2 and earlier) to 256 MiB (Vivado 2018.3) in the Xilinx HBM IP, Vivado projects cannot be successfully generated.
- Added a workaround to the **debug-*.tcl** scripts of the **hbm_test** and **host_hbm** example FPGA designs for a JTAG clock frequency issue where the default JTAG clock frequency of 15 MHz used prevents the HBM showing up correctly in Vivado 2018.3 Hardware Manager.

Revision History

Date	Revision	Nature of change
11th December 2018	1.0	Initial release 0.1.0 for ES1 (only).
24th January 2019	1.1	Fixes to example FPGA designs for compatibility with Vivado 2018.3.