Getting Started with the SDAccel Environment on Nimbix Cloud
# Revision History

The following table shows the revision history for this document.

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>08/16/2017</td>
<td>2017.2</td>
<td>Changes are: Modified Login URL.</td>
</tr>
<tr>
<td>06/20/2017</td>
<td>2017.1</td>
<td>Changes are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated for 2017.1 release.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Changed Nimbix to Material Compute flow.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated figures throughout.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Added reference to the Xilinx VU9P Developer Board DSA file.</td>
</tr>
<tr>
<td>12/15/2016</td>
<td>2016.3</td>
<td>Updated supported board configurations</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated commands throughout</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Updated Figure 15.</td>
</tr>
<tr>
<td>11/14/2016</td>
<td>2016.2</td>
<td>Initial Xilinx release.</td>
</tr>
</tbody>
</table>
Table of Contents

Revision History .................................................................................................................................................. 2
Chapter 1: SDAccel Environment on Nimbix Cloud Overview ............................................................................. 4
  Introduction .................................................................................................................................................. 4
Chapter 2: Nimbix Account Creation, Login, and Setup ..................................................................................... 5
  Creating an Account .................................................................................................................................... 5
  Logging into Your Nimbix Account .............................................................................................................. 6
  Transferring Data to and from Nimbix ......................................................................................................... 7
    /data Directory: Important Notes ............................................................................................................. 9
Chapter 3: Using the SDAccel Environment in Nimbix Cloud ............................................................................ 10
  Introduction ................................................................................................................................................. 10
  Step 1: Login to your Nimbix account ...................................................................................................... 10
  Step 2: Load SDAccel Examples from the GitHub Site ............................................................................. 12
  Step 3: Compile the Design ....................................................................................................................... 15
  Step 4: Terminate the Session ................................................................................................................... 18
  Step 5: Deploy the Application ................................................................................................................ 20
Appendix A: Legal Notices .......................................................................................................................... 23
  Please Read: Important Legal Notices .................................................................................................... 23
Chapter 1: SDAccel Environment on Nimbiax Cloud Overview

Introduction

This document demonstrates how to get started using SDAccel™ Environment in the Nimbiax Cloud.

The SDAccel Environment, a member of the SDx™ family of development environments for systems and software engineers, enables up to 25X better performance/watt for data center application acceleration leveraging FPGAs.

Xilinx® has partnered with Nimbiax Inc®. to deliver an opportunity to test drive SDAccel and see how FPGA-based acceleration can speed-up your OpenCL C, C/C++ and RTL Kernels. The Nimbiax Accelerated Compute Cloud provides a purpose-built compute cloud for big data and computation. It now hosts the SDAccel development environment for companies who want to streamline evaluations and accelerate FPGA OpenCL kernels.

You can use multiple approaches when working with Nimbiax Cloud, such as:

- Using Nimbiax to compile and deploy your application with the Xilinx FPGA-based boards hosted on Nimbiax
- Compiling the application on your local machine, transferring the compiled files to Nimbiax, and then deploying the application on Nimbiax

This document describes how to use both approaches and contains a step-by-step tutorial to help familiarize you with using SDAccel under Nimbiax Environment.

**IMPORTANT:** When compiling the application to be deployed on Xilinx FPGAs based boards hosted by Nimbiax, you need specify one of the following board configurations:

<table>
<thead>
<tr>
<th>FPGA Board</th>
<th>Device Support Archive (DSA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha Data V7</td>
<td>xilinx:adm-pcie-7v3:1ddr:3.0</td>
</tr>
<tr>
<td>Alpha Data KU3</td>
<td>xilinx:adm-pcie-ku3:2ddr-xpr:4.0</td>
</tr>
<tr>
<td>Xilinx KU115 Developer Board</td>
<td>xilinx:xil-accel-rd-pcie3-ku115:4ddr:4.0</td>
</tr>
<tr>
<td>Xilinx VU9P Developer Board</td>
<td>xilinx:xil-accel-rd-vu9p:4ddr-xpr:4.1</td>
</tr>
</tbody>
</table>
Chapter 2: Nimbix Account Creation, Login, and Setup

Creating an Account

To create an account, do the following:

1. Register for the SDAccel® Environment at the Nimbix website. Navigate with your browser to https://www.nimbix.net/xilinx.
   The following figure shows the Login page.

   ![Login Page](image_url)

   **Figure 1: Nimbix Cloud Landing Page**

2. Complete the registration form.
   Upon approval of your account, you receive an email confirmation along with information introducing you to the SDAccel Environment.
Logging into Your Nimbix Account

To login to your account:


![Nimbix Login Page](image)

   **Figure 2: Nimbix Login Page**

2. Enter your user name and a password and then click **Log In** to login into your account.

   When you login to your account, the Nimbix interface **Compute** opens.

![Dashboard Page](image)

   **Figure 3: Dashboard Page**
Chapter 2: Nimbix Account Creation, Login, and Setup

The **Dashboard** provides you information about:

- The jobs you are currently running and their status
- Previously run jobs

The **Compute** page allows you to compile and deploy your applications.

---

**Transferring Data to and from Nimbix**

When working with SDAccel, you might need to transfer files from your machine. Nimbix. SFTP is the preferred file transfer method. The following instructions demonstrate file transfer using the FileZilla SFTP client software.

*Note: Follow this [link](#) for more information regarding file transfers.*

---

1. Download and install FileZilla.
2. Open FileZilla and configure the following:
   a. In **Host**: Specify *drop.jarvice.com*
   b. In **Port**: Specify the value *22*
   c. In **Username**: Specify your Nimbix Username
   d. The **Password**: Specify your JARVICE API key

   To obtain your API key, login to your Nimbix account, go to your account settings and copy the API Key from the Settings page, as shown in Figure 5.
3. Select the **Quickconnect** button to connect to Nimbix.

   When the connection establishes successfully, you see the `/data` directory on the remote site, as shown in the following figure:

   ![Remote Site data Directory](image)

   **Figure 5: API Key on the Settings Page**

Using FileZilla interface, you can now start to transfer files to and from Nimbix.

---

**IMPORTANT:** If you work behind a company firewall and are unable to connect to Nimbix, speak with your IT department and request the access to the 22 (TCP) port.
/data Directory: Important Notes

Working in the Nimbix Environment, you will need to use the /data directory in different situations, so it is important to understand the following features:

- Data located in the Nimbix /data directory is preserved after you logout from your Nimbix account. Copy critical files to the /data directory if you want to preserve them for later reuse. You can also create sub-directories under /data if necessary.

- The /data directory is where you need to exchange data between Nimbix and your local machine. For example, if you need code located in your local machine for SDAccel compilation on Nimbix, transfer this file to the /data directory first and then copy from /data to the final folder.

- The /data directory is the location from which to deploy compiled applications on the FPGA boards hosted by Nimbix. Copy all necessary files to the /data directory first.

Chapter 3: Using the SDAccel Environment in Nimbix Cloud describes these activities.
Chapter 3: Using the SDAccel Environment in Nimbix Cloud

Introduction

The tutorial uses the `vadd` example available in the `/getting_started/misc` folder at this location:

https://github.com/Xilinx/SDAccel_Examples/.

Step 1: Login to your Nimbix account


![Figure 7: Nimbix Account Login](image)

2. Provide your Username and Password and select Log In.

3. In the opened page, select Compute, shown in the following figure, to open the Compute page.

![Figure 8: Compute](image)
Chapter 3: Using the SDAccel Environment in Nimbix Cloud

The **Compute** page opens, as shown in the following figure:

![Figure 9: The Compute Page](image)

a. Use **SDAccel Development** to:
   - Compile and run your application in CPU and Hardware Emulation modes.
   - Compile your application and transfer the compiled design to the `/data` directory for Hardware deployment using FPGA boards hosted by Nimbix.

b. Use **SDAccel Runtime** to:
   - Deploy your application on the FPGA Boards hosted by Nimbix. For application deployment, the compiled application and data files should be located in the `/data` directory.
Step 2: Load SDAccel Examples from the GitHub Site

1. Launch SDAccel Development Environment
2. On the **Compute** page, select **SDAccel Development**.
3. Select the **Gui workflow**, shown in the following figure:

   ![Figure 10: Selecting Gui](image)

   **Figure 10: Selecting Gui**

4. The **Builder** page opens. You can change default options, such as the machine configuration to use, window size, and so forth, if necessary. The following figure shows the top-level GUI page. When your configurations are set, continue the job submission by pressing **Submit**.

   ![Figure 11: Task Builder](image)

   **Figure 11: Task Builder**

   This automatically makes the **Dashboard** page active, as shown in **Figure 12**, where you can see that you have a new active job.
5. Click in the desktop preview space to open a new tab with an interactive desktop session.

   **Note:** *It can take several seconds before the* desktop preview opens.

   The ready-to-use Linux desktop environment becomes available, as shown in the following figure:

   **Figure 13: Ready-to-Use Linux Desktop**
Two icons are on the desktop:

- **Terminal**: Opens a new terminal with a ready-to-use SDAccel environment.
- **data**: The directory in which you transfer the compiled applications for hardware deployment. See /data Directory: Important Notes for more information about the /data directory.

**Download the Latest SDAccel Examples from GitHub**

1. Open the **Terminal**.

2. In the **Terminal**, enter the following command to get the latest version of the SDAccel examples hosted on GitHub:

   ```
   git clone https://github.com/Xilinx/SDAccel_Examples /data/examples
   ```

   After the command executes, the terminal should look like the following figure:

   ![Terminal with Latest Version of SDAccel Examples](image)

   **Figure 14**: Terminal with Latest Version of SDAccel Examples
Step 3: Compile the Design

This tutorial uses a simple “vadd” example.

In the opened Terminal, go to the `/data/examples/getting_started/misc/vadd` directory by typing the following command:

```bash
cd /data/examples/getting_started/misc/vadd
```

### Compile and Run the Application in CPU Emulation Mode

The CPU Emulation mode verifies functional correctness of the application. To compile the vadd example for a CPU Emulation mode, do the following:

1. Run the following command in an opened Terminal from the `vadd` directory:

   ```bash
   make TARGETS=hw_emu all
   ```

   This command generates:
   
   - A host executable, named `vadd` in the current directory.
   - A `krnl_vadd.sw_emu.xilinx_xil-accel-rd-ku115_4ddr-xpr.xclbin file` (XCLBIN) located in the `xclbin` directory.

2. Before running the application in CPU Emulation mode, run the following commands:

   ```bash
   export LD_LIBRARY_PATH=$XILINX_SDACCEL/runtime/lib/x86_64/:$LD_LIBRARY_PATH
   export XCL_EMULATION_MODE=sw_emu
   ```

   These commands set the library path, the `XCL_EMULATION_MODE` variable, and the DSA file.

3. Now you can launch the application using the following command:

   ```bash
   ./vadd
   ```

   **Note:** This application uses the `XCL_EMULATION_MODE` variable and discovered emulated devices to select the correct XCLBIN file.

4. Upon successful execution of the application, the following output displays:

   ```text
   Result Match: i = 1021 CPU result = 3063 Krnl Result = 3063
   Result Match: i = 1022 CPU result = 3066 Krnl Result = 3066
   Result Match: i = 1023 CPU result = 3069 Krnl Result = 3069
   TEST PASSED
   ```
Compile and Run the Application in Hardware Emulation Mode

The Hardware Emulation mode checks the correctness of the logic generated for the custom compute units, and estimates the application performance. To compile the example for a Hardware Emulation mode:

1. Run the following command in an opened Terminal from the vadd directory:
   
   ```
   make TARGETS=hw_emu all
   ```
   
   This command generates:
   
   - A host executable, named `vadd` in the current directory.
   - A `krnl_vadd.hw_emu.xilinx_xil-accel-rd-kul15_4ddr-xpr.xclbin` (XCLBIN) file located in the `xclbin` directory.

   **IMPORTANT:** If you:
   
   - Continue to use the same Terminal you used to run CPU Emulation mode, then before running the application in a Hardware Emulation mode, run the following command:
     
     ```
     export XCL_EMULATION_MODE=hw_emu
     ```
   
   - Compiled the application for Hardware Emulation mode using a different terminal, run the following commands first:
     
     ```
     export LD_LIBRARY_PATH=$XILINX_SDACCEL/runtime/lib/x86_64/:$LD_LIBRARY_PATH
     export XCL_EMULATION_MODE=hw_emu
     ```

2. Now you can launch the application using the following command:
   
   ```
   ./vadd
   ```

   **Note:** This application uses the `XCL_EMULATION_MODE` variable and discovered emulated devices to select the correct XCLBIN file.

3. Upon successful execution of the application, the following output displays:
   
   ```
   ...
   Result Match: i = 1021 CPU result = 3063 Krnl Result = 3063
   Result Match: i = 1022 CPU result = 3066 Krnl Result = 3066
   Result Match: i = 1023 CPU result = 3069 Krnl Result = 3069
   TEST PASSED
   ```

4. After the Hardware emulation has been completed, you can explore the application performance by opening the `sdaccel_profile_summary.html` file in the Firefox Web Browser using the following command:
   
   ```
   firefox sdaccel_profile_summary.html
   ```

   You should see the report shown in Figure 15.
SDAccel Profile Summary

Generated on: 2017-05-31 16:40:45
Profiled application: vadd
Target platform: Xilinx
Target devices: xilinx:xil-accel-rd-ku115:4ddr-xpr:4.0-0
Flow mode: Hardware Emulation
Tool version: 2017.1

OpenCL API Calls

<table>
<thead>
<tr>
<th>API Name</th>
<th>Number Of Calls</th>
<th>Total Time (ms)</th>
<th>Minimum Time (ms)</th>
<th>Average Time (ms)</th>
<th>Maximum Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>clReleaseProgram</td>
<td>1</td>
<td>8214.62</td>
<td>8214.62</td>
<td>8214.62</td>
<td>8214.62</td>
</tr>
<tr>
<td>clFinish</td>
<td>1</td>
<td>1789.25</td>
<td>1789.25</td>
<td>1789.25</td>
<td>1789.25</td>
</tr>
<tr>
<td>clCreateProgramWithBinary</td>
<td>1</td>
<td>385.065</td>
<td>385.065</td>
<td>385.065</td>
<td>385.065</td>
</tr>
<tr>
<td>clEnqueueMigrateMemObjects</td>
<td>2</td>
<td>0.417502</td>
<td>0.144475</td>
<td>0.208751</td>
<td>0.273627</td>
</tr>
<tr>
<td>clCreateBuffer</td>
<td>3</td>
<td>0.36518</td>
<td>0.107924</td>
<td>0.120393</td>
<td>0.140637</td>
</tr>
<tr>
<td>clEnqueueTask</td>
<td>1</td>
<td>0.370325</td>
<td>0.370325</td>
<td>0.370325</td>
<td>0.370325</td>
</tr>
<tr>
<td>clCreateKernel</td>
<td>1</td>
<td>0.021707</td>
<td>0.021707</td>
<td>0.021707</td>
<td>0.021707</td>
</tr>
<tr>
<td>clReleaseKernel</td>
<td>1</td>
<td>0.015673</td>
<td>0.015673</td>
<td>0.015673</td>
<td>0.015673</td>
</tr>
<tr>
<td>clSetKernelArg</td>
<td>4</td>
<td>0.012121</td>
<td>0.001052</td>
<td>0.00303025</td>
<td>0.00627</td>
</tr>
<tr>
<td>clGetExtensionFunctionAddress</td>
<td>1</td>
<td>0.01140</td>
<td>0.01140</td>
<td>0.01140</td>
<td>0.01140</td>
</tr>
<tr>
<td>clCreateCommandQueue</td>
<td>1</td>
<td>0.008386</td>
<td>0.008386</td>
<td>0.008386</td>
<td>0.008386</td>
</tr>
<tr>
<td>clRetainMemObject</td>
<td>3</td>
<td>0.007995</td>
<td>0.000993</td>
<td>0.00266533</td>
<td>0.005062</td>
</tr>
<tr>
<td>clGetPlatformInfo</td>
<td>4</td>
<td>0.006882</td>
<td>0.000704</td>
<td>0.0017205</td>
<td>0.003184</td>
</tr>
<tr>
<td>clReleaseMemObject</td>
<td>6</td>
<td>0.000719</td>
<td>0.000314</td>
<td>0.00111983</td>
<td>0.004954</td>
</tr>
<tr>
<td>clCreateContext</td>
<td>1</td>
<td>0.006136</td>
<td>0.006136</td>
<td>0.006136</td>
<td>0.006136</td>
</tr>
<tr>
<td>clReleaseCommandQueue</td>
<td>1</td>
<td>0.005685</td>
<td>0.005685</td>
<td>0.005685</td>
<td>0.005685</td>
</tr>
<tr>
<td>clGetDeviceIDs</td>
<td>2</td>
<td>0.004723</td>
<td>0.000737</td>
<td>0.0023615</td>
<td>0.003566</td>
</tr>
<tr>
<td>clRetainDevice</td>
<td>2</td>
<td>0.003288</td>
<td>0.001195</td>
<td>0.001644</td>
<td>0.002093</td>
</tr>
<tr>
<td>clGetDeviceInfo</td>
<td>2</td>
<td>0.002944</td>
<td>0.001101</td>
<td>0.001472</td>
<td>0.001643</td>
</tr>
<tr>
<td>clReleaseDevice</td>
<td>2</td>
<td>0.00179</td>
<td>0.000298</td>
<td>0.000895</td>
<td>0.001492</td>
</tr>
<tr>
<td>clReleaseContext</td>
<td>1</td>
<td>0.001713</td>
<td>0.001713</td>
<td>0.001713</td>
<td>0.001713</td>
</tr>
</tbody>
</table>

Figure 15: SDAccel Profile Summary Report
Compile and Prepare the Application for Hardware Deployment

1. To compile the example for hardware deployment, run the following command in an opened Terminal from the getting_started/vadd directory:
   
   ```
   make TARGETS=hw all
   ```
   
   This command generates:
   
   - A host executable named `vadd` in the current directory.
   - A `krnl_vadd.hw.xilinx_xil-accel-rd-ku115_4ddr-xpr.xclbin` (XCLBIN) file located in the `xclbin` directory.

2. To simplify the application deployment on Nimbix, the host’s executable should have the `.exe` file extension. Copy the `vadd` executable to `vadd.exe` with the following command:
   
   ```
   mv vadd vadd.exe
   ```

   **IMPORTANT:** At this step, you can close the current SDAccel Development session; however, all files **not** located in the `/data` directory will be lost. Copy any files you wish to preserve to the `/data` directory before you continue.

Step 4: Terminate the Session

To terminate the current session:

1. From the Dashboard, select shutdown.

   ![Figure 16: Shutdown Option](image)

2. Select Yes.

   You should see in the Dashboard page that the process status is **Completed**. You should also receive a corresponding E-mail.

   ![Figure 17: Instance Status Check: Completed](image)
WARNING! After terminating the session, you might see that the job is still *Processing*, as shown in the following figure.

3. If the job is still processing, check **Force Stop**, select **Yes**, then select **Shutdown** to terminate the job.

![Figure 18: Instance Status Check: Processing](image_url)

![Figure 19: Instance Management Buttons](image_url)
Step 5: Deploy the Application

The application is now compiled and all the necessary files for hardware deployment are located in the /data directory. To deploy the application:

1. From the Compute page of the Nimbix Environment, select SDAccel Runtime > batch, shown in the following figure.

![Figure 20: Selecting Batch Workflow](image1)

2. On the opened page:
   
   a. Select the target platform from the Select Machine drop-down menu.
      
      For this example, select KU115 FPGA:

      ![Figure 21: Selecting the Board](image2)

   b. Select the vadd.exe host executable file located in the /examples/getting_started/misc/vadd/ directory from the Executable dialog box, shown in the following figure.

      ![Figure 22: Selecting the Host Executable File](image3)

The application automatically detects the XCLBIN file for deployment; no further arguments need to be specified.

![Figure 23: Arguments Field](image4)

**Note:** The directory where the host executable file is located is considered a current directory by Nimbix.
Chapter 3: Using the SDAccel Environment in Nimbix Cloud

c. To obtain a graphical Trace report, check one of the following options:

![Timeline trace](image1)

![Device profiling](image2)

Figure 24: TRACE Report Options

d. Select **Submit** to launch the deployment, and obtain the results.

If the execution of the application is successful, the following status displays:

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Start Date</th>
<th>Compute Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDAccel Runtime:batch</td>
<td><strong>Completed</strong></td>
<td>2016-11-03 20:36:43</td>
<td>00:00:20</td>
</tr>
</tbody>
</table>

Figure 25: Successful Application Execution Status

Nimbix sends you the application log file by email automatically.

e. Select the **SDAccel Runtime: batch** job to expand it, and then select **Output** to see the log file generated by the application.

You should see log messages similar to the CPU and Hardware Emulation modes, as shown in Figure 26.
Figure 26: Job Output

In addition to the email with the log file you receive from Nimbix, you can save the log file on your local machine by clicking the **Download** button in the output windows, as shown in the following figure:

Figure 27: Download Log File

The application automatically generates the performance report in the `sdaccel_profile_summary.html` file, located in the `xbinst` directory. To explore this report you can either:

- Restart the SDAccel Environment and open this file using the Firefox Web Browser.
- Copy the file to your local machine using the method described in Transferring Data to and from Nimbix and then opening it using your Web Browser.
Appendix A: Legal Notices

Please Read: Important Legal Notices

The information disclosed to you hereunder (the “Materials”) is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available “AS IS” and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx's limited warranty, please refer to Xilinx's Terms of Sale which can be viewed at https://www.xilinx.com/legal.htm#tos; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx's Terms of Sale which can be viewed at https://www.xilinx.com/legal.htm#tos.

AUTOMOTIVE APPLICATIONS DISCLAIMER

AUTOMOTIVE PRODUCTS (IDENTIFIED AS “XA” IN THE PART NUMBER) ARE NOT WARRANTED FOR USE IN THE DEPLOYMENT OF AIRBAGS OR FOR USE IN APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE (“SAFETY APPLICATION”) UNLESS THERE IS A SAFETY CONCEPT OR REDUNDANCY FEATURE CONSISTENT WITH THE ISO 26262 AUTOMOTIVE SAFETY STANDARD (“SAFETY DESIGN”). CUSTOMER SHALL, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE PRODUCTS, THOROUGHLY TEST SUCH SYSTEMS FOR SAFETY PURPOSES. USE OF PRODUCTS IN A SAFETY APPLICATION WITHOUT A SAFETY DESIGN IS FULLY AT THE RISK OF CUSTOMER, SUBJECT ONLY TO APPLICABLE LAWS AND REGULATIONS GOVERNING LIMITATIONS ON PRODUCT LIABILITY.

© Copyright 2017 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.