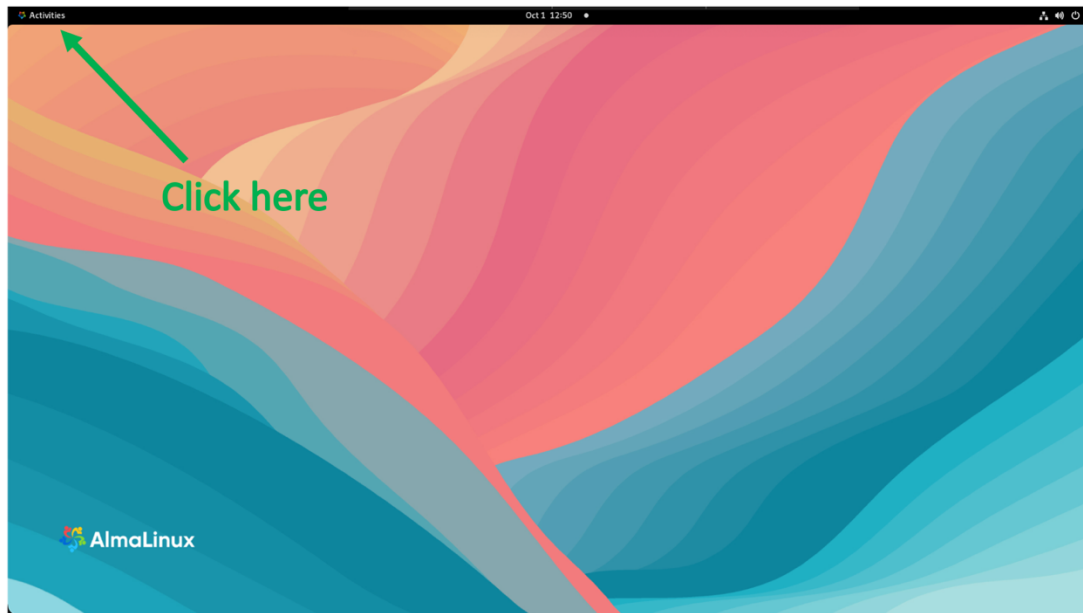


# Programming the FPGAs

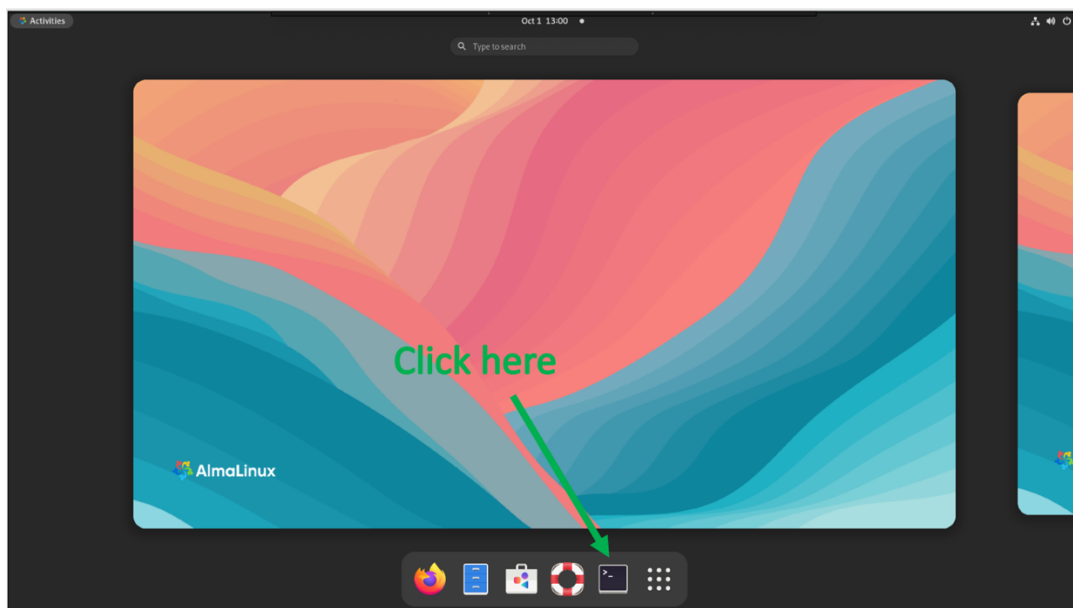
This guide details how to program the Apollo FPGAs using the Vivado GUI.

## 1) Open a Terminal

Once logged onto a Cornell computer connected to an Apollo board, open a terminal. **(For production testing, the node to log onto is `lnx231.classe.cornell.edu`.)** One way this can be done is by clicking the “Activities” button in the top left-hand corner of the screen.



Next, click the terminal icon now at the bottom of the screen.



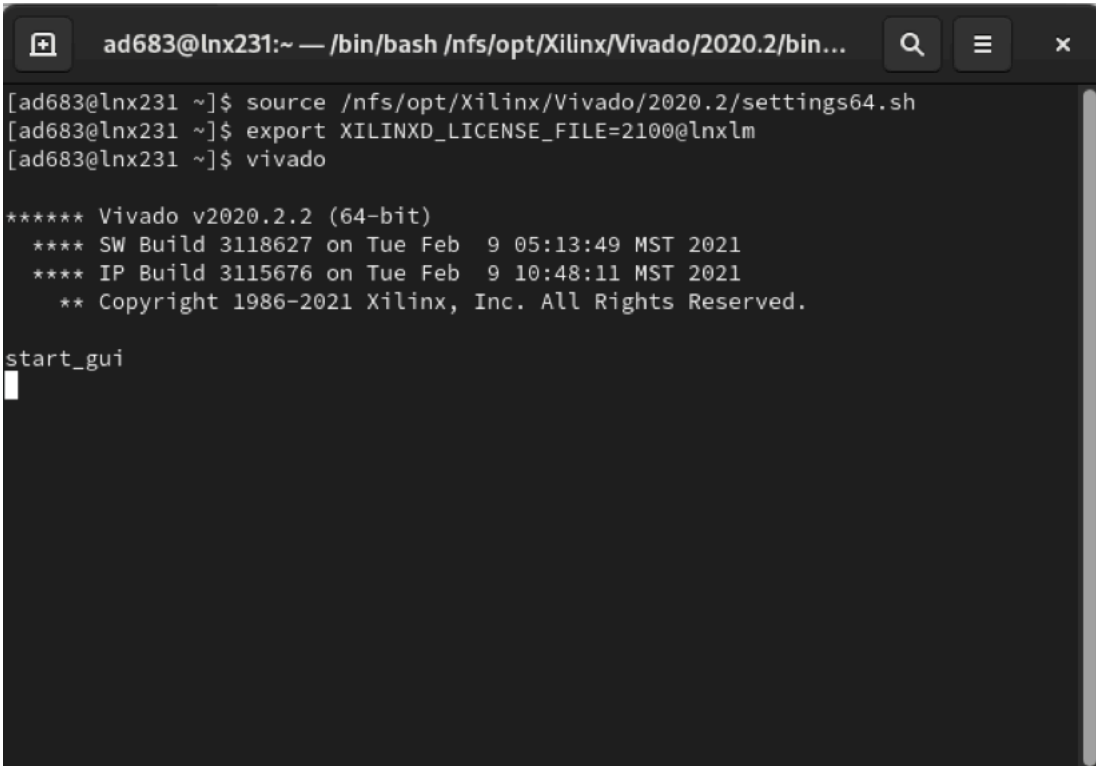
## 2) Open the Vivado GUI

We are going to use this terminal to open an instance of the Vivado GUI. Before we can do that, we must tell the computer where Vivado is installed and which version we want to use. Do this by entering the following commands into the terminal:

```
source /nfs/opt/Xilinx/Vivado/2020.2/settings64.sh
export XILINXD_LICENSE_FILE=2100@lnxlm
vivado
```

*Note that vivado creates a log file where it is launched, so make sure you have write permission in the directory you are writing in.*

Upon entering the above commands into the terminal, the terminal should print the following messages:

A screenshot of a terminal window with a dark background. The title bar at the top shows the user 'ad683@lnx231' and the current directory path. The terminal displays the execution of three commands: 'source /nfs/opt/Xilinx/Vivado/2020.2/settings64.sh', 'export XILINXD\_LICENSE\_FILE=2100@lnxlm', and 'vivado'. Following these commands, Vivado prints its version and build information, including the software and IP build numbers, dates, and times, along with the copyright notice for Xilinx, Inc. The prompt then changes to 'start\_gui' with a cursor on the next line.

```
ad683@lnx231:~ — /bin/bash /nfs/opt/Xilinx/Vivado/2020.2/bin...
[ad683@lnx231 ~]$ source /nfs/opt/Xilinx/Vivado/2020.2/settings64.sh
[ad683@lnx231 ~]$ export XILINXD_LICENSE_FILE=2100@lnxlm
[ad683@lnx231 ~]$ vivado

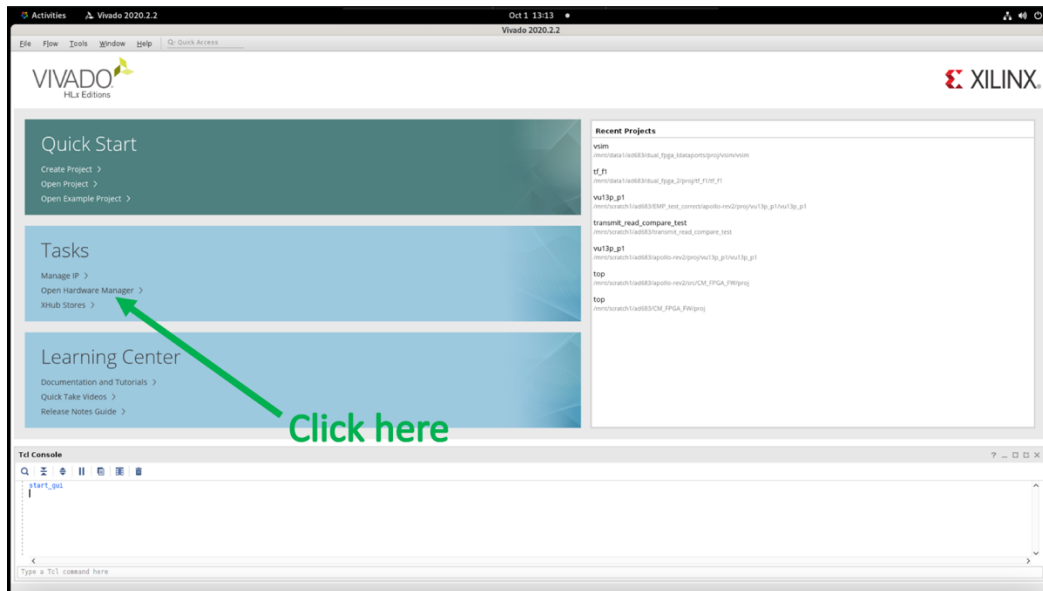
***** Vivado v2020.2.2 (64-bit)
**** SW Build 3118627 on Tue Feb  9 05:13:49 MST 2021
**** IP Build 3115676 on Tue Feb  9 10:48:11 MST 2021
** Copyright 1986-2021 Xilinx, Inc. All Rights Reserved.

start_gui
█
```

After a few seconds, the Vivado GUI should appear on your screen.

### 3) Open the Vivado Hardware Manager

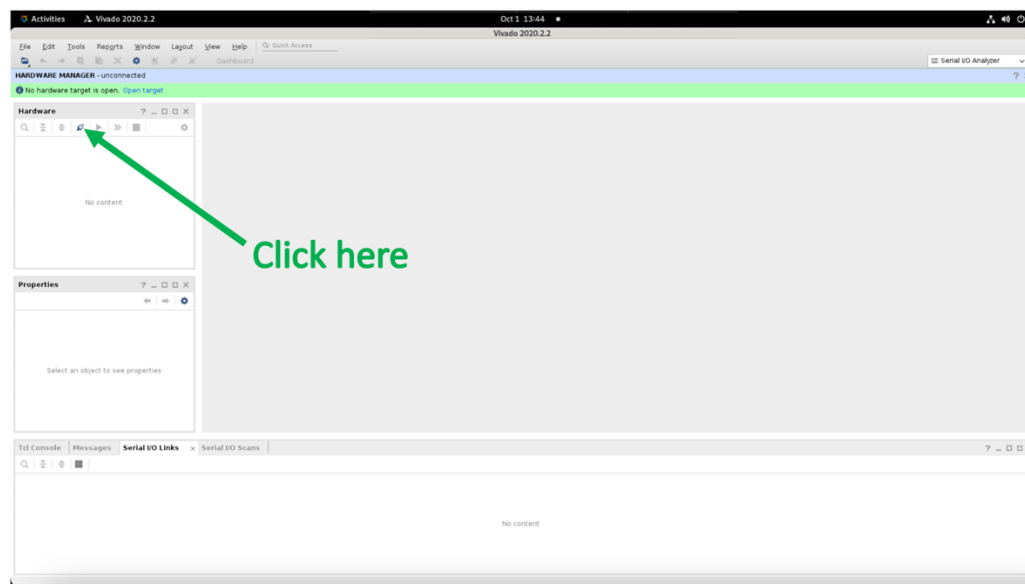
Next, click “Open Hardware Manager” in the “Tasks” section.



This should open the hardware manager designed to program, test, and interact in other ways with the Apollo board.

### 4) Connect to the Hardware Target

Before we do any of those things, we need to connect the board to the hardware manager. Look in the box labeled “Hardware” in the top left of the screen. Click the auto-connect button that looks like a dark wrench and a light wrench facing each other.



Auto-connect should take a few seconds to complete. Upon completion, the hardware manager should have connected to a target named: "xilinx\_tcf/Xilinx/00001631afcb01"

You should see this target name next to a green square with a red dot in the "Hardware" box like so. Below that, you should see the two devices corresponding to the two FPGAs. They are represented by jagged blue squares. The name next to one device should read "xcvu13p\_0", and the other should read "xcvu13p\_1". If instead you see a long scrollable list of other components, don't worry. This just means that at least one of the boards is already programmed. The device may read as "Programmed" or "Not programmed" next to the name to indicate this. Either setting is fine. The procedure for programming the boards will be the same. The important point is that you can find the two devices of interest in that list. However, you may want to click the downward facing arrow next to programmed devices to hide the MGT list so you don't have to scroll around to find the devices we want to program.

**Hardware**

Make sure this hardware target matches what you see

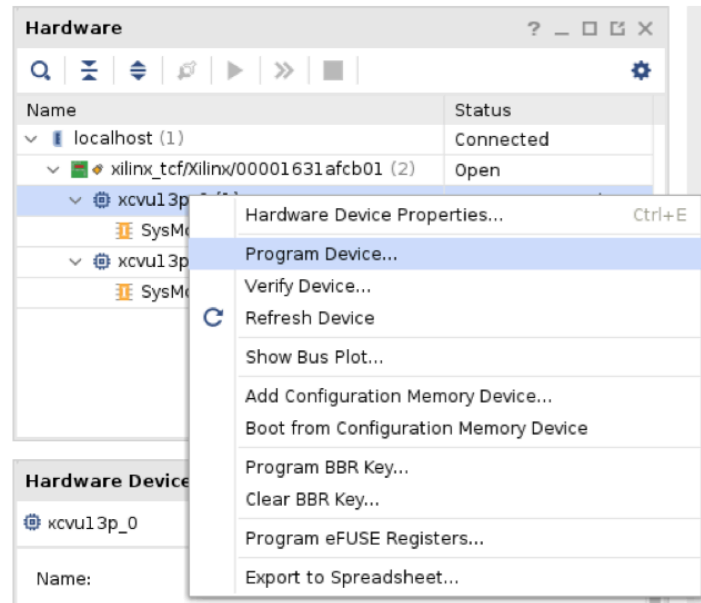
Name	Status
localhost (1)	Connected
xilinx_tcf/Xilinx/00001631afcb01 (2)	Open
xcvu13p_0 (1)	Not programmed
SysMon (System Monitor)	
xcvu13p_1 (1)	Not programmed
SysMon (System Monitor)	

Devices we will program

You might see a long list of Quads, MGTs, or other info here if the FPGAs are programmed, that is ok we will still reprogram them

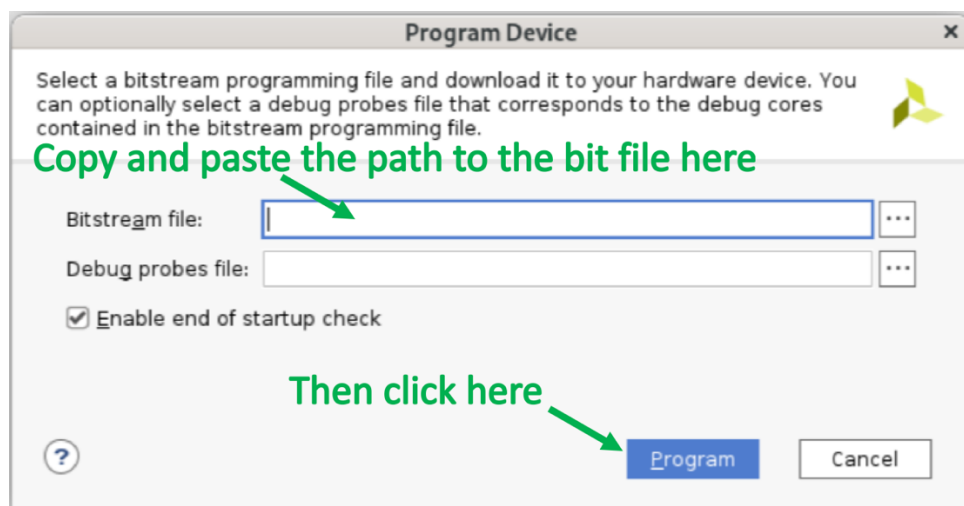
### 5) Select FPGA to Program

Next, right click the first device (the one labeled “xcvu13p\_0”) to open an options menu shown below. Then click “Program Device...”

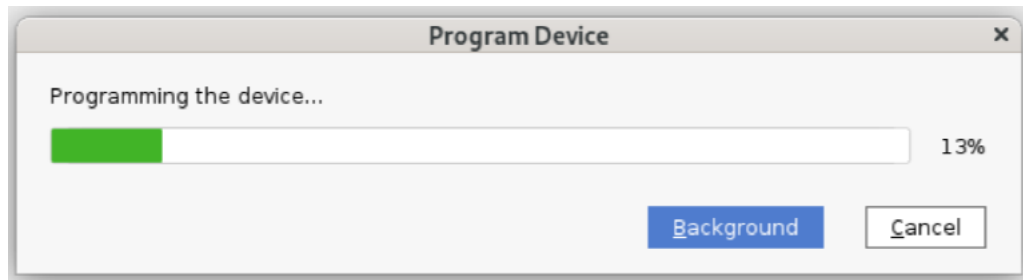


### 6) Choose the Firmware with which to Program the FPGA

That will open a window titled “Program Device” as shown below. Finally, copy and paste the path to your bit file (e.g. /nfs/cms/tracktrigger/apollo/clocktest/clock\_test.bit) in the “Bitstream file:” text box. Make sure you are copying the correct firmware. Some firmwares are specific to one FPGA or the other. Other firmwares expect certain connectors on the external links and using the wrong firmware could damage the board as a result. Finally, click the blue “Program” button to program the first FPGA (xcvu13p\_0) with the desired firmware.



You will then see a loading screen like the one below. It may take a few minutes to finish, but upon completion the FPGA will be programmed.



#### 7) Program the other FPGA

Programming the second FPGA (xcvu13p\_1) is a similar procedure. Repeat the procedure detailed in steps 5) and 6). This time right clicking on the other device for step 5) and select "Program Device...". Then repeat step 6) making sure to copy and paste the correct firmware for FPGA 2 (xcvu13p\_1).