Configuring Custom HDL Models using Simulink

Prerequisites:

Recommended versions: **Vivado 2021.1** – **Matlab R2021B_U4**

Recommended terminal for Windows: Cygwin

Make sure that the **Vitis 2021.1** is installed.

The latest branch:

https://github.com/analogdevicesinc/HighSpeedConverterToolbox/tree/cn0585_v1

1. Instructions to build the toolbox from terminal:

>git clone https://github.com/analogdevicesinc/HighSpeedConverterToolbox.git

> cd HighSpeedConverterToolbox

../HighSpeedConverterToolbox> git submodule update --init --recursive

../HighSpeedConverterToolbox > git checkout cn0585_v1

To avoid tool mismatches, before opening MATLAB set this variable in the terminal:

../HighSpeedConverterToolbox> export ADI_IGNORE_VERSION_CHECK=TRUE

../HighSpeedConverterToolbox > cd CI/scripts

../HighSpeedConverterToolbox/CI/scripts > make build HDLBRANCH=cn0585_v1

This will also make a clone of the HDL repo and checkout the desired branch.

In Matlab current folder list select navigate to the folder where the files had been copied from previous step.

Launch MATLAB in the root of the HighSpeedConverterToolbox folder:

../HighSpeedConverterToolbox/CI/scripts > cd ../../

../HighSpeedConverterToolbox > matlab .

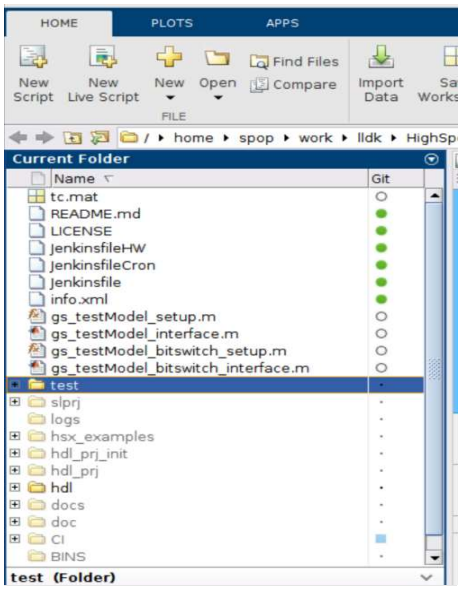

Figure 1. High Speed Converter Toolbox Sources

right click on test -> Add to Path -> Selected folders and subfolders

right click on hdl -> Add to Path -> Selected folders and subfolders

In the Matlab command window set the path to Vivaldo installation folder. The tool path should be replaced with the user's Vivado path.

e.g. **hdlsetuptoolpath('ToolName', 'Xilinx Vivado', 'ToolPath', '</opt/Xilinx/Vivado/2021.1/bin/vivado>')**

Expand the test folder and double click on the desired Simulink test model, as shown in Figure 2.

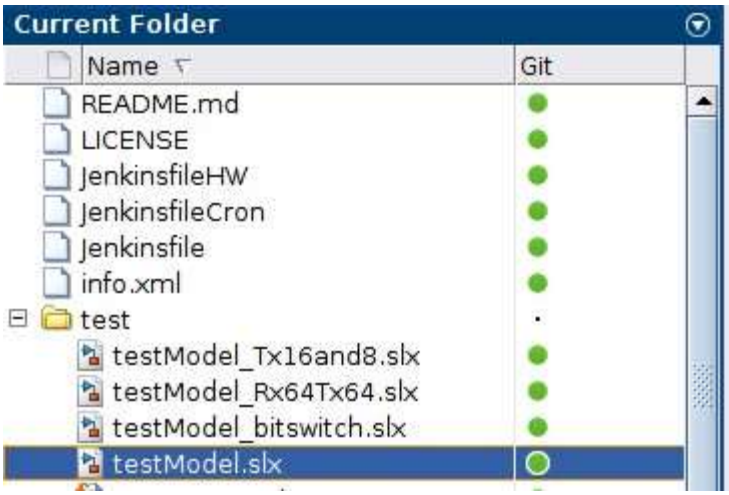

Figure 2. Simulink Test Model

After opening the Simulink model, right click on the HDL_DUT and launch the HDL Workflow Advisor as shown in Figure 3, and Figure 4.

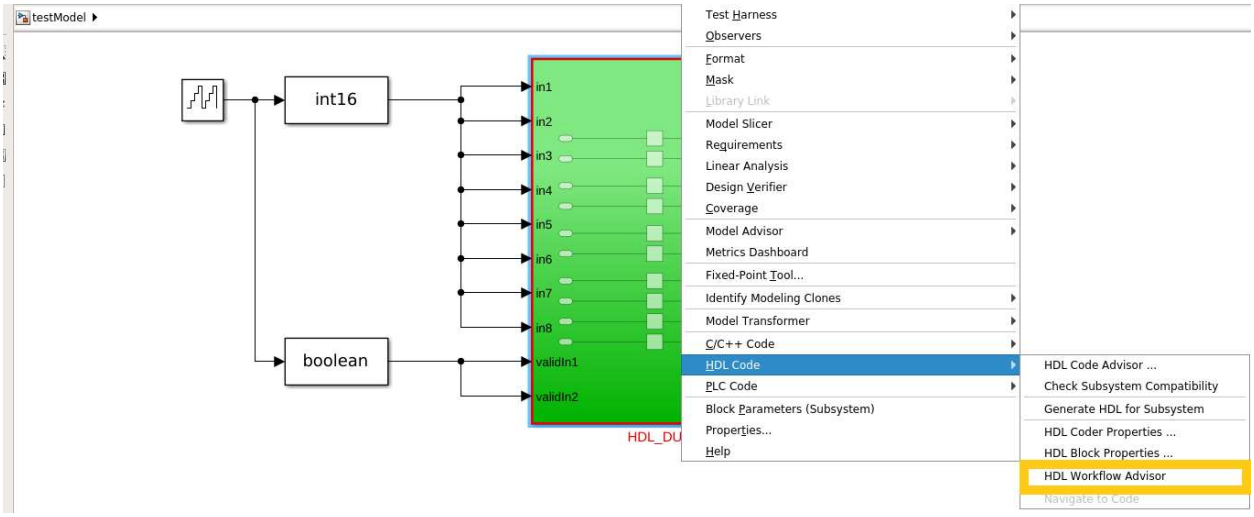

Figure 3. HDL Workflow Advisor Launching

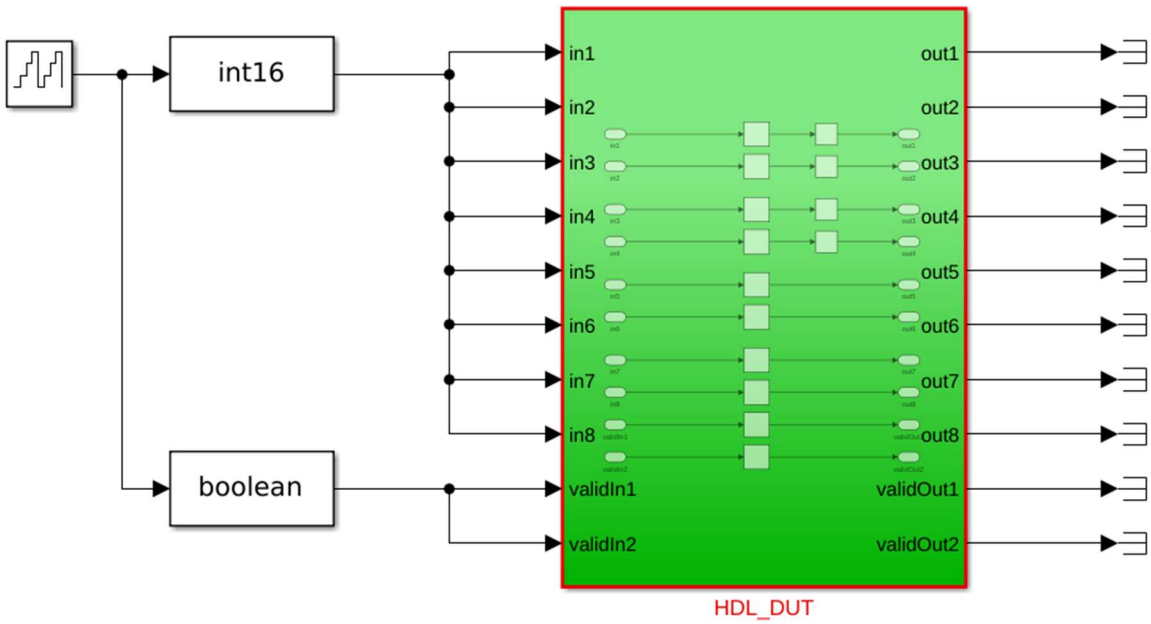

Figure 4. Simulink Device Under Test

Close this expected warning that will appear, as shown in Figure 5.

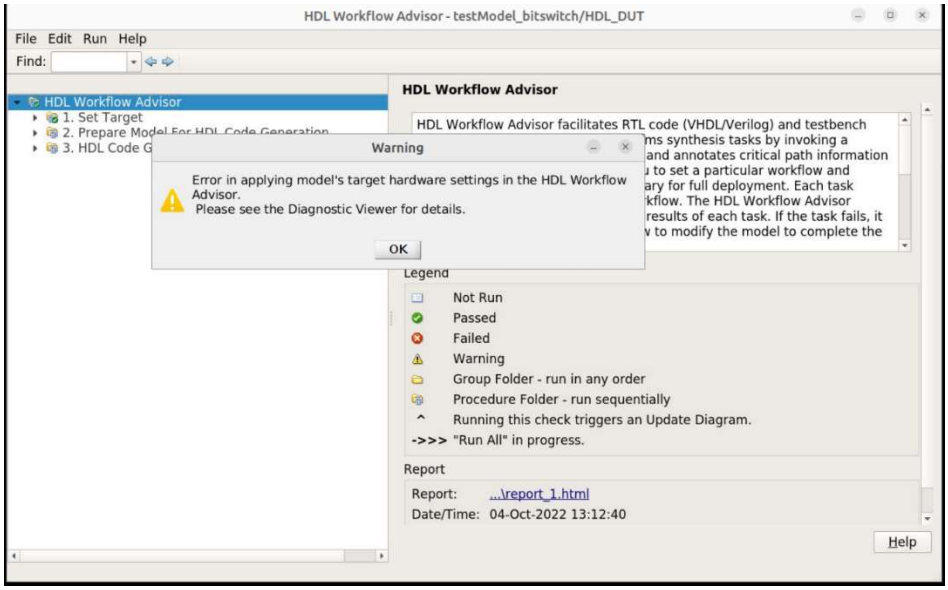

Figure 5. Expected HDL Workflow Advisor Warning

1.1 Select IP Core Generation, choose the desired project and carrier from the dropdown list and check the Allow unsupported version box. Change the project folder name if desired. Finally press the Run this Task button.

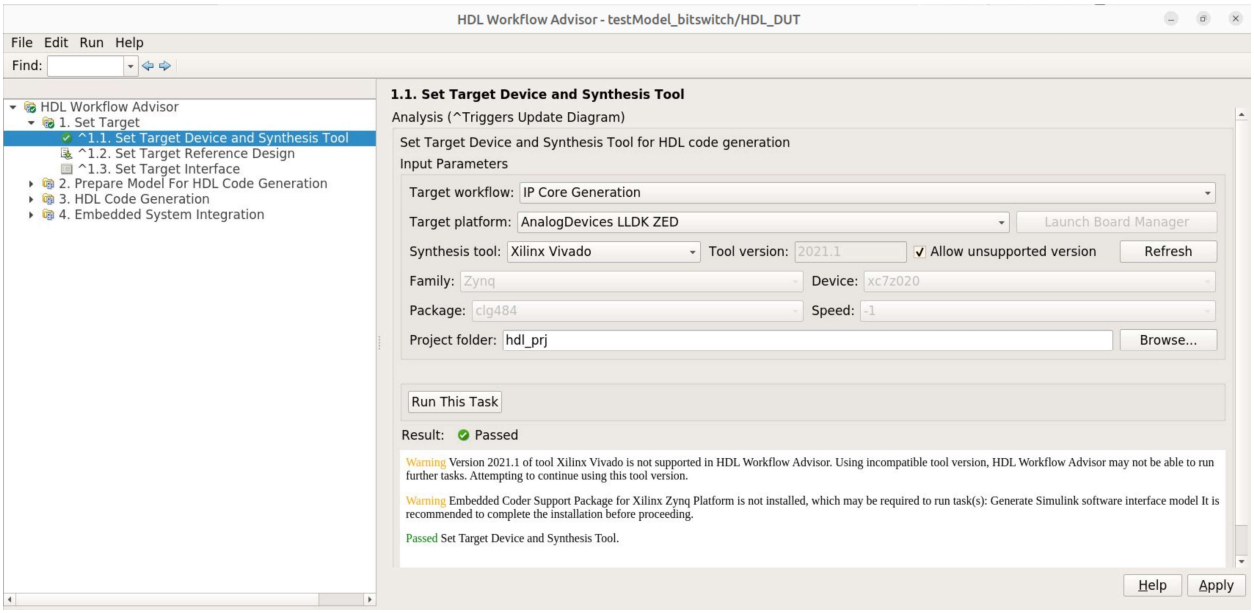

Figure 6. Set Target Device and Synthesis Tool

1.2 Choose the RX, TX, or RX&TX configuration, then run task:

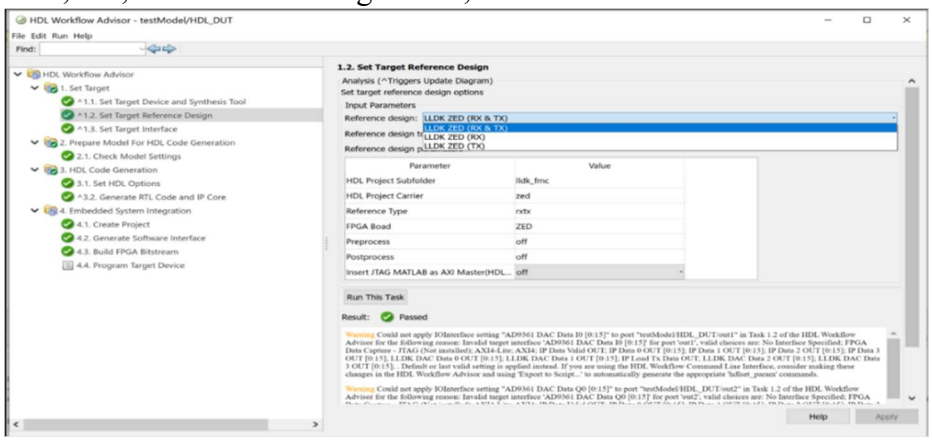

Figure 7. Set Target Reference Design

The HDL DUT in Rx &Tx configuration will pe placed between the Tx and the Rx path, as shown in Figure 8.

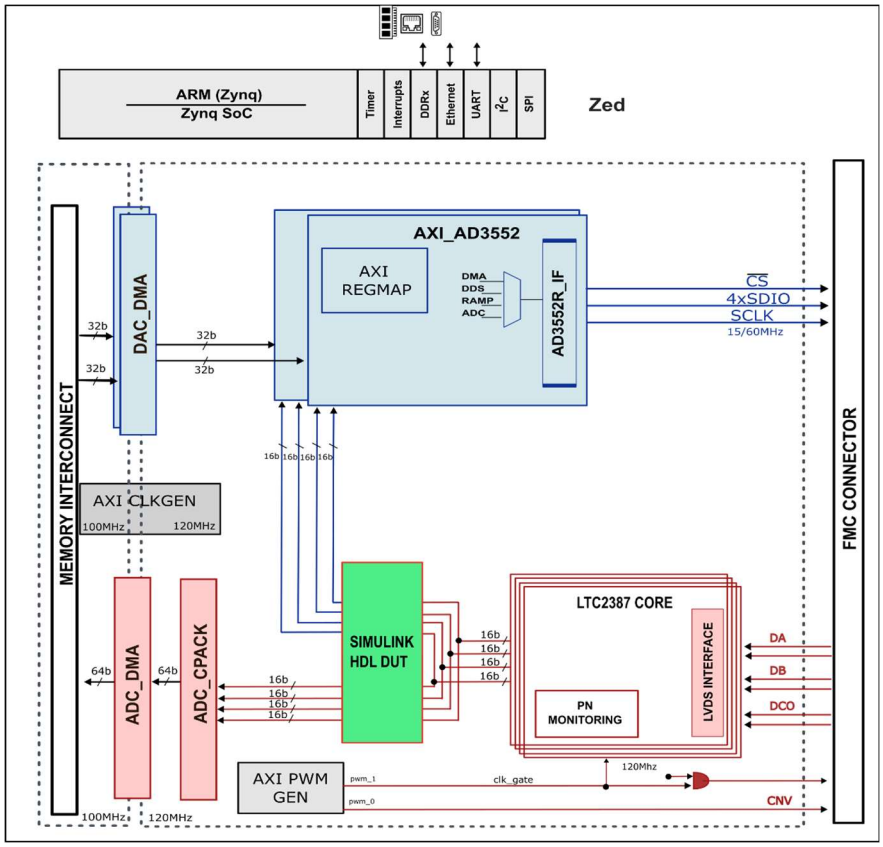

Figure 8. HDL Block Design After Inserting the Simulink Block in Rx & Tx Configuration

The HDL DUT in Tx configuration will pe placed between the Tx and the Rx path, as shown in Figure 9.

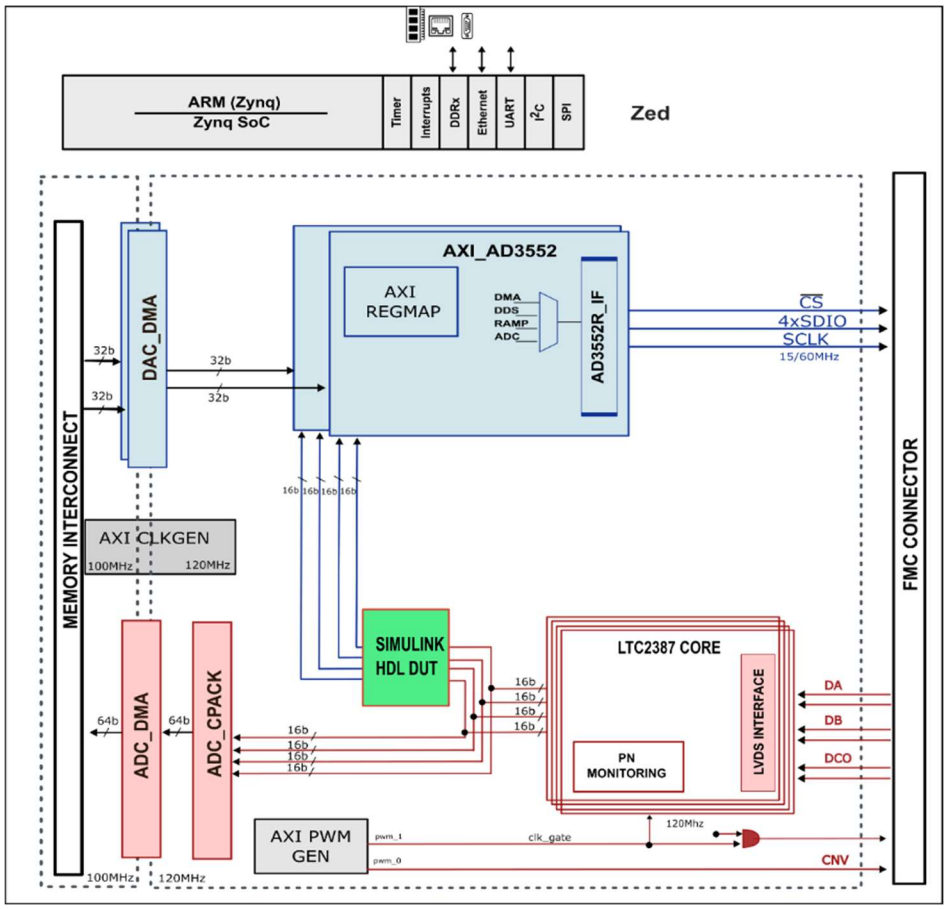

Figure 9. HDL Block Design After Inserting the Simulink Block in Tx Configuration

The HDL DUT in Rx configuration will pe placed between the Tx and the Rx path, as shown in Figure 10.

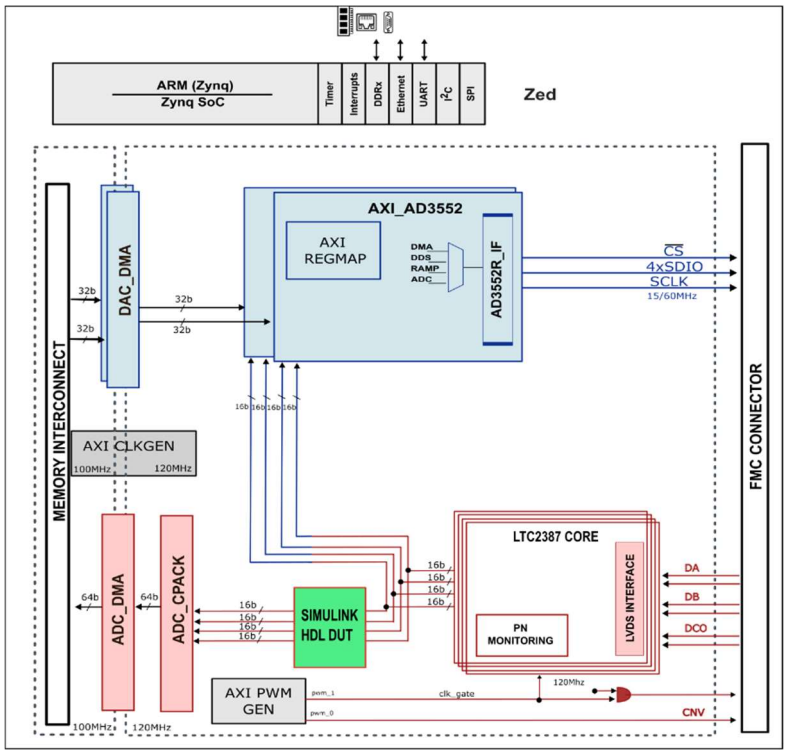

Figure 10. HDL Block Design After Inserting the Simulink Block in Rx Configuration

1.3 At this point you need to assign every port, as shown in Figure 11 and Figure 12, and run the task.

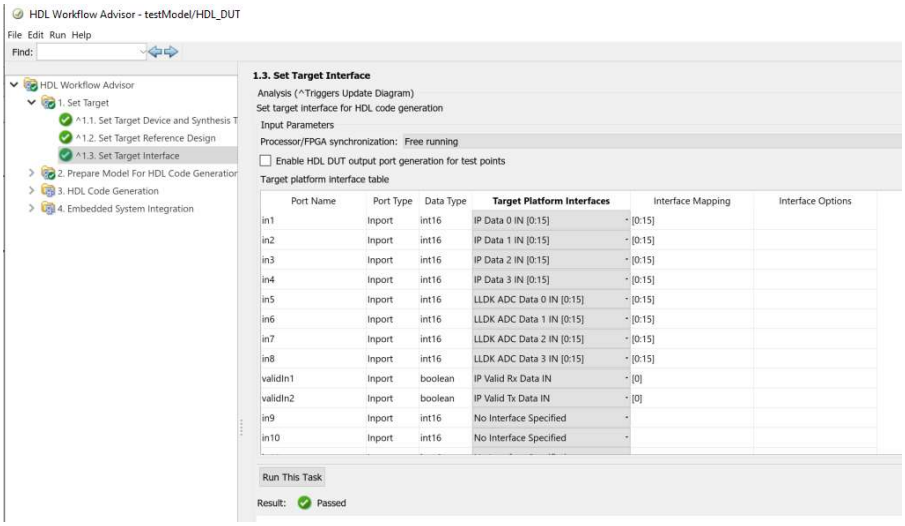

Figure 11. Set Target Interface Input Ports (Rx & Tx Reference Design)

The LLDK ADC DATA <x> IN is the data in offset binary format captured by the ADC interface IP. IP sends the data at a variable sample rate (default is 15MHz but can be changed using the IIO Oscilloscope/ Python) along with the validIn<x> signal which has the logic value 1 for a clock period (8.33ns) when the data has changed.

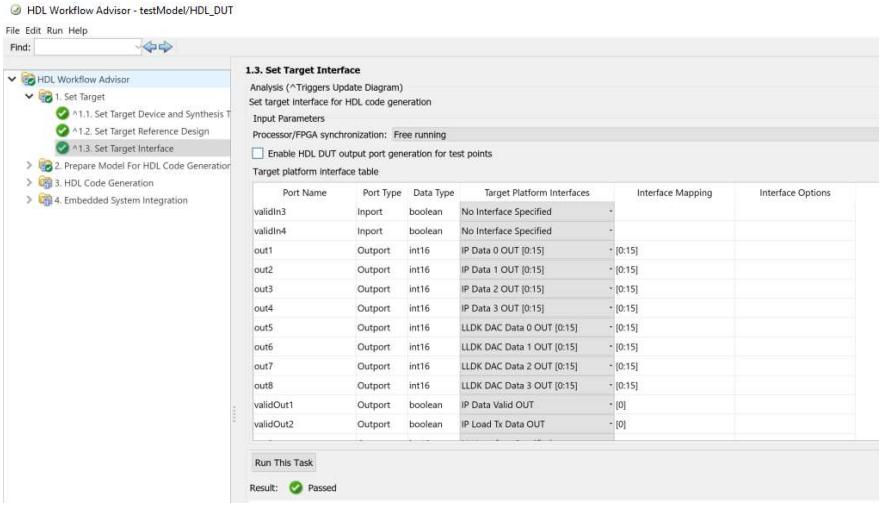

Figure 12. Set Target Interface Output Ports (Rx & Tx Reference Design)

IP DATA <x> OUT is the data in offset binary format sent to the DAC interface IP. Data must be sent at 15MSPS when both channels are enabled or at 30MSPS when only one channel is enabled. The validOut<x> signal should have the same behavior as validIn. If you make changes to the data captured by the adc (delay for 1 clock period) and want to send it to the dac output, make sure you delay the validOut signal at the same time. If the feedback resistors are placed in the default position, which is +/-10V, a 0000h code will represent -10.382V and a ffffh code will represent 10.380V as described in *Table 1*.

	Hardware				AD3552R Register Settings		
	Output Span	**V_{ZS} (V)**	**V_{FS} (V)**	**Feedback Connection**	**CH0_CH1_OUTPUT_RANGE**	**CHx_GAIN_SCALING_N**	**CHx_OFFSET**
CH0	+/- 10V (Default)	-10.382	10.380	DAC0 to DAC0_RFB0	0x100	0	-245
	+/- 5V	-5.165	5.166	DAC0 to DAC0_RFB0_X2	0x011	0	-495
	10V	-0.165	10.163	DAC0 to DAC0_RFB0_X2	0x010	0	495
	5V	-0.078	5.077	DAC0 to DAC0_RFB0_X1	0x001	0	0
	2.5V	-0.198	2.701	DAC0 to DAC0_RFB0_X1	0x000	3	-48
CH1	+/- 10V (Default)	-10.382	10.380	DAC1 to DAC1_RFB1	0x100	0	-245
	+/- 5V	-5.165	5.166	DAC1 to DAC1_RFB1_X2	0x011	0	-495
	10V	-0.165	10.163	DAC0 to DAC0_RFB0_X2	0x010	0	495
	5V	-0.078	5.077	DAC1 to DAC1_RFB1_X1	0x001	0	0
	2.5V	-0.198	2.701	DAC1 to DAC1_RFB1_X1	0x000	3	-48

Table 1: AD3552R DAC Output Span Configuration

Table 2, Table 3 and Table 4 show port descriptions for HDL DUT in Tx, Rx and Tx & Rx Reference Designs.

Interface signal name	Width	Description
IP Data 0 IN	16	Custom IP data input signal. This signal is connected to the AXI_LTC2387_0 IP ADC DATA port in the ADI reference design.
IP Data 1 IN	16	Custom IP data input signal. This signal is connected to the AXI_LTC2387_1 IP ADC DATA port in the ADI reference design.
IP Data 2 IN	16	Custom IP data input signal. This signal is connected to the AXI_LTC2387_2 IP ADC DATA port in the ADI reference design.
IP Data 3 IN	16	Custom IP data input signal. This signal is connected to the AXI_LTC2387_3 IP ADC DATA port in the ADI reference design.
IP Valid Tx Data IN	1	Input signal that has logic 1 value for a clock cycle period when the data starts to be valid.
CN0585 DAC Data 0 OUT	16	AD3552R_0 DAC 0 channel data. To be used as input into the AD3552R interface IP.
CN0585 DAC Data 1 OUT	16	AD3552R_0 DAC 1 channel data. To be used as input into the AD3552R interface IP.
CN0585 DAC Data 2 OUT	16	AD3552R_1 DAC 0 channel data. To be used as input into the AD3552R interface IP.
CN0585 DAC Data 3 OUT	16	AD3552R_1 DAC 1 channel data. To be used as input into the AD3552R interface IP.
IP Load Tx Data OUT	1	Custom IP output signal used to notify the design that the IP is ready to receive new input data. Output signal that must be logic '1' for a clock cycle period when the data starts to be valid.

Table 2: HDL DUT Ports for Transmit Reference Design (Tx)

Interface signal name	Width	Description
CN0585 ADC Data 0 IN	16	Custom IP data input signal. This signal is connected to the AXI_LTC2387_0 IP ADC DATA port in the ADI reference design.
CN0585 ADC Data 1 IN	16	Custom IP data input signal. This signal is connected to the AXI_LTC2387_1 IP ADC DATA port in the ADI reference design.
CN0585 ADC Data 2 IN	16	Custom IP data input signal. This signal is connected to the AXI_LTC2387_2 IP ADC DATA port in the ADI reference design.
CN0585 ADC Data 3 IN	16	Custom IP data input signal. This signal is connected to the AXI_LTC2387_3 IP ADC DATA port in the ADI reference design.
IP Valid Rx Data IN	1	Input signal that has logic 1 value for a clock cycle period when the data starts to be valid.
IP Data 0 OUT	16	ADAQ23876 ADC 0 channel data. To be used as input for the ADC CPAK IP.
IP Data 1 OUT	16	ADAQ23876 ADC 1 channel data. To be used as input for the ADC CPAK IP.
IP Data 2 OUT	16	ADAQ23876 ADC 2 channel data. To be used as input for the ADC CPAK IP.
IP Data 3 OUT	16	ADAQ23876 ADC 3 channel data. To be used as input for the ADC CPAK IP.
IP Data Valid OUT	1	Output signal that has to be logic '1' for a clock cycle period when the data starts to be valid.

Table 3: HDL DUT Ports for Receive Reference Design (Rx)

Interface signal name	Width	Description
IP Data 0 IN	16	Custom IP data input signal. This signal is connected to the AXI_LTC2387_0 IP ADC DATA port in the ADI reference design.
IP Data 1 IN	16	Custom IP data input signal. This signal is connected to the AXI_LTC2387_1 IP ADC DATA port in the ADI reference design.
IP Data 2 IN	16	Custom IP data input signal. This signal is connected to the AXI_LTC2387_2 IP ADC DATA port in the ADI reference design.
IP Data 3 IN	16	Custom IP data input signal. This signal is connected to the AXI_LTC2387_3 IP ADC DATA port in the ADI reference design.
CN0585 ADC Data 0 IN	16	Custom IP data input signal. This signal is connected to the AXI_LTC2387_0 IP ADC DATA port in the ADI reference design.
CN0585 ADC Data 1 IN	16	Custom IP data input signal. This signal is connected to the AXI_LTC2387_1 IP ADC DATA port in the ADI reference design.
CN0585 ADC Data 2 IN	16	Custom IP data input signal. This signal is connected to the AXI_LTC2387_2 IP ADC DATA port in the ADI reference design.
CN0585 ADC Data 3 IN	16	Custom IP data input signal. This signal is connected to the AXI_LTC2387_3 IP ADC DATA port in the ADI reference design.
IP Valid Rx Data IN	1	Input signal that has logic 1 value for a clock cycle period when the data starts to be valid.
IP Valid Tx Data IN	1	Input signal that has logic 1 value for a clock cycle period when the data starts to be valid.
IP Data 0 OUT	16	ADAQ23876 ADC 0 channel data. To be used as input for the ADC CPAK IP.
IP Data 1 OUT	16	ADAQ23876 ADC 1 channel data. To be used as input for the ADC CPAK IP.
IP Data 2 OUT	16	ADAQ23876 ADC 2 channel data. To be used as input for the ADC CPAK IP.
IP Data 3 OUT	16	ADAQ23876 ADC 3 channel data. To be used as input for the ADC CPAK IP.
CN0585 DAC Data 0 OUT	16	AD3552R_0 DAC 0 channel data. To be used as input into the AD3552R interface IP.
CN0585 DAC Data 1 OUT	16	AD3552R_0 DAC 1 channel data. To be used as input into the AD3552R interface IP.
CN0585 DAC Data 2 OUT	16	AD3552R_1 DAC 0 channel data. To be used as input into the AD3552R interface IP.
CN0585 DAC Data 3 OUT	16	AD3552R_1 DAC 1 channel data. To be used as input into the AD3552R interface IP.
IP Data Valid OUT	1	Output signal that has to be logic '1' for a clock cycle period when the data starts to be valid.
IP Load Tx Data OUT	1	Custom IP output signal used to notify the design that the IP is ready to receive new input data. The duration must be 1 clock cycle.

Table 4: HDL DUT Ports for Transmit & Receive Reference Design (Tx & Rx)

2.1 Run the task, as shown in Figure 13.

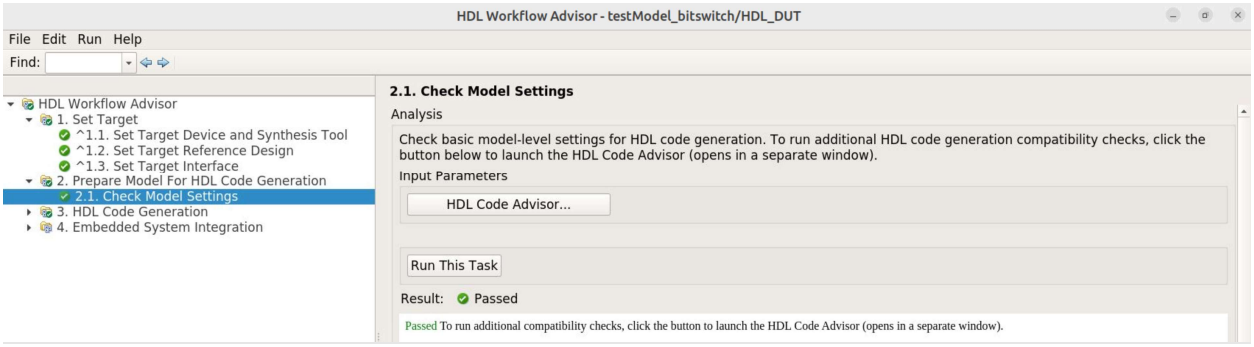

Figure 13. Check Model Settings

3.1 Select Verilog for the HDL Code Generation Settings, then run task as shown in Figure 14.

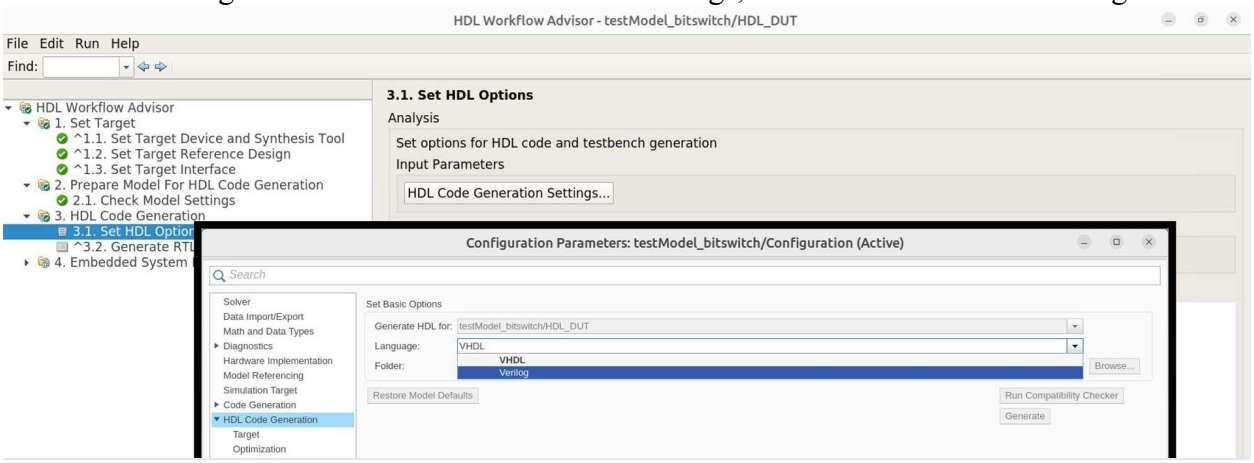

Figure 14. Set HDL Options

3.2 Run the task, as shown in Figure 15.

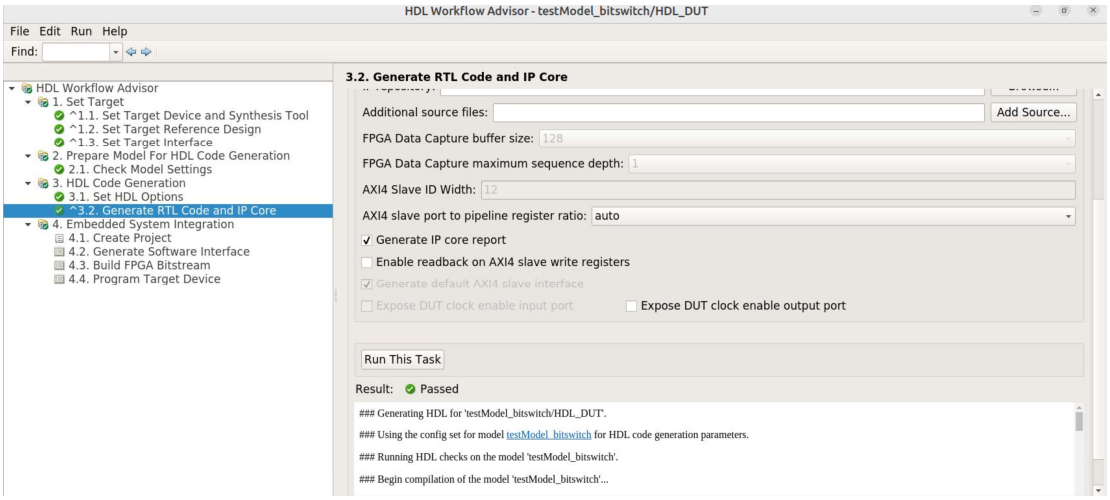

Figure 15. Generate RTL Code and IP Core.

4.1 Run the task (this will create the Vivado block design in the hdl_prj/vivado_ip_prj folder, or the project folder name that was chosen in 1.1), as shown in Figure 16.

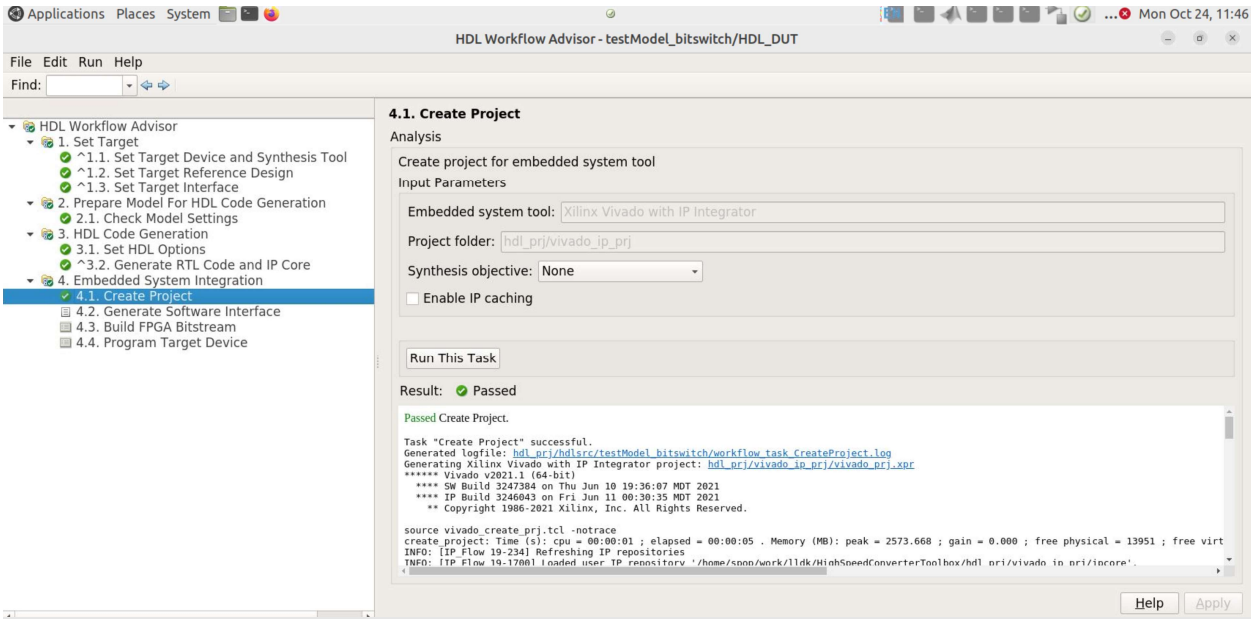

Figure 16. Create Project.

4.2 Run the task in Figure 17.

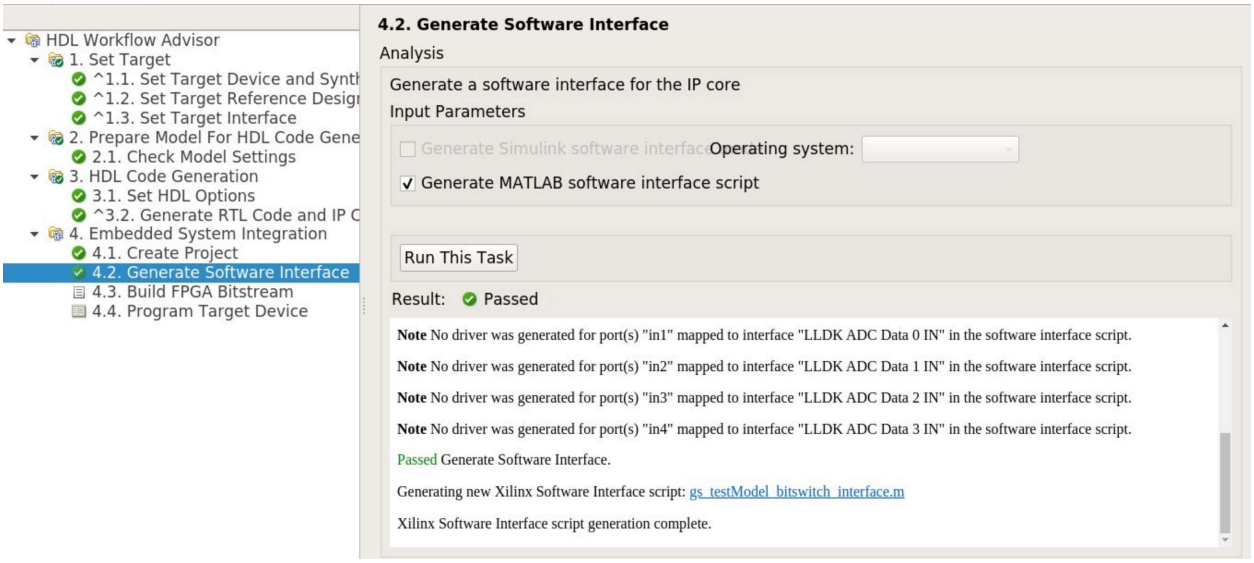

Figure 17. Generate Software Interface

4.3 Choose the "Custom" option for the Tcl file synthesis build, then Browse for the adi_build.tcl file located under HighSpeedConverterToolbox/CI/scripts, as shown in Figure 18:

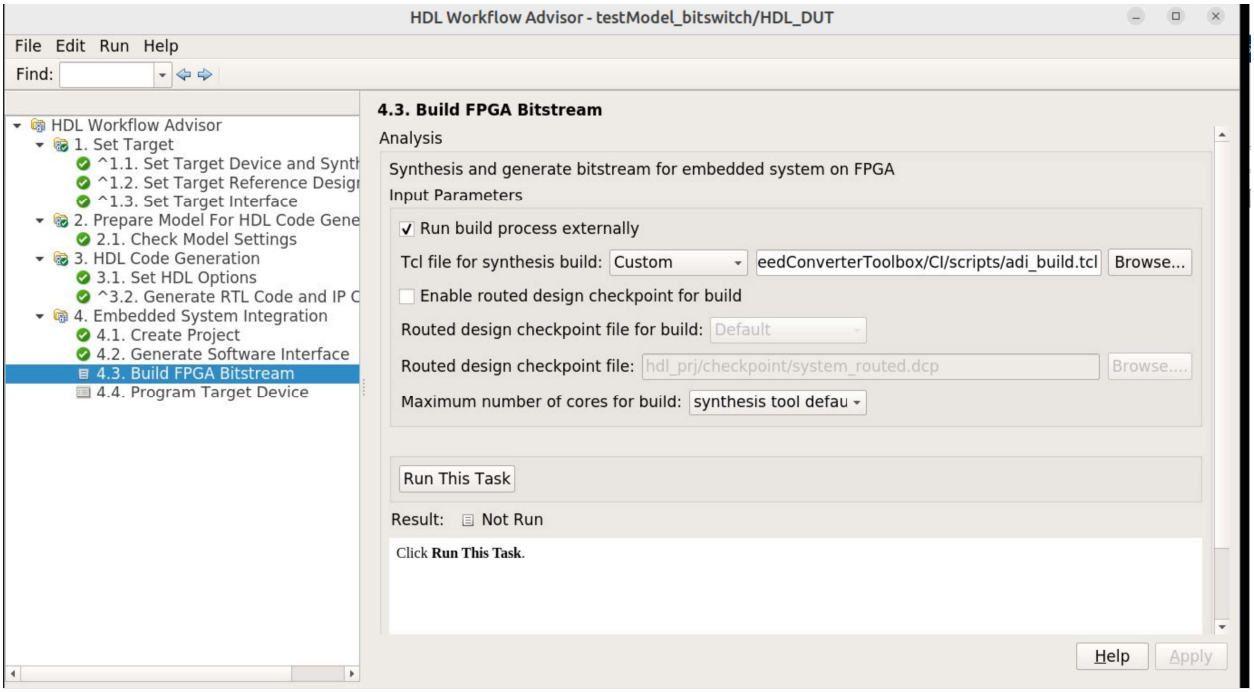

Figure 18. Build FPGA Bitstream

A bash prompt will open, and you can see the entire build process log file, as shown in Figure 19, and Figure 20.

This step usually takes about an hour.

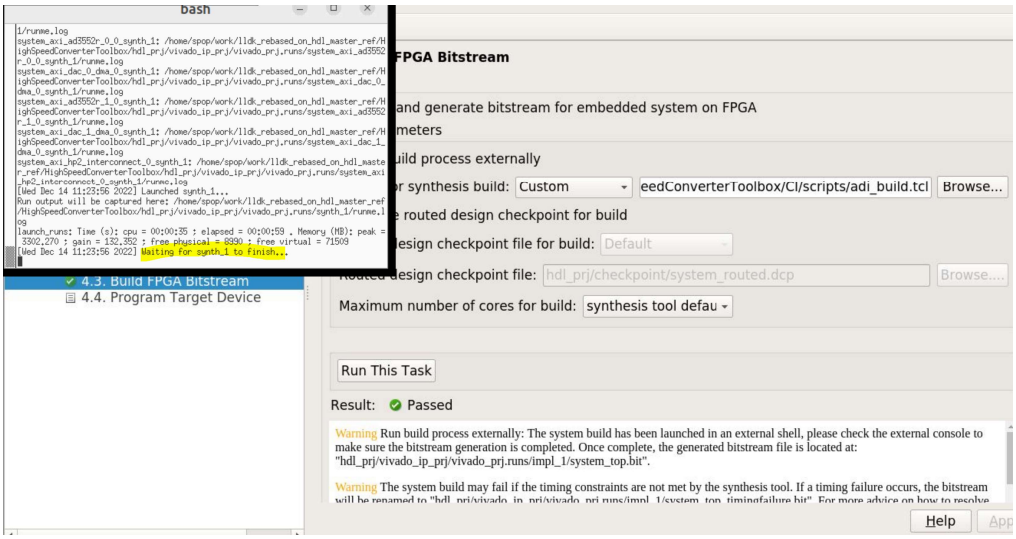

Figure 19. Build FPGA Bitstream Console Output

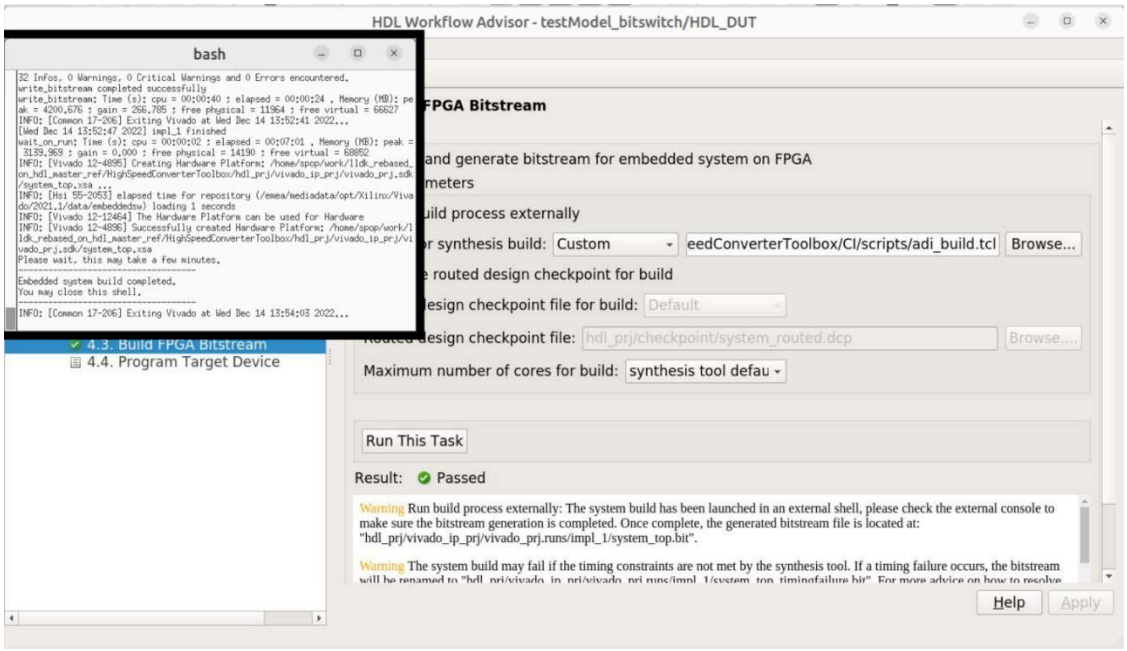

Figure 20. Build FPGA Bitstream Task Complete Message

In the end you will get this message, and the generated BOOT.BIN file will be located here, Figure 21.

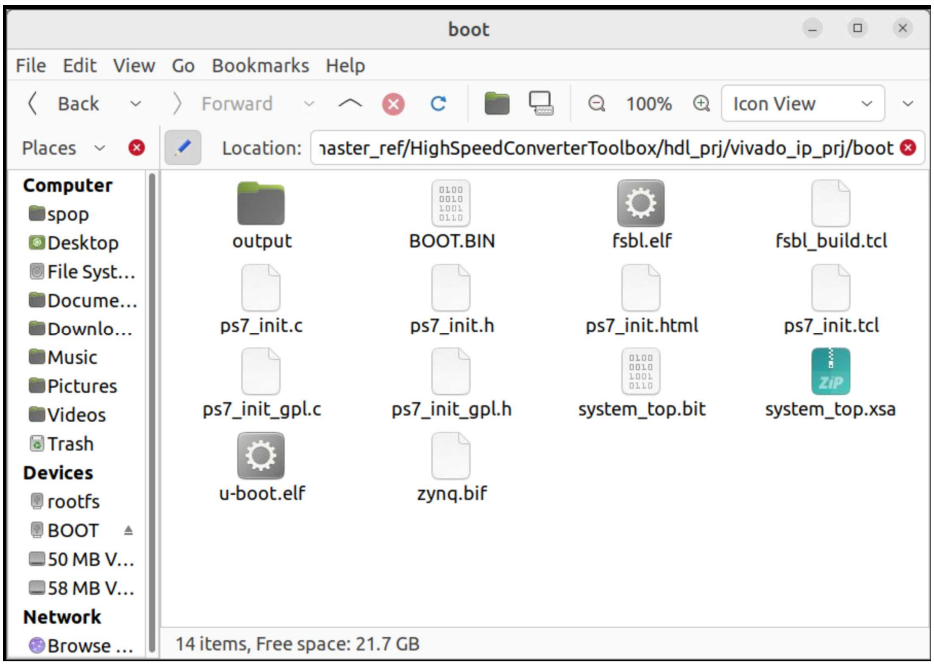

Figure 21. Location of the BOOT.BIN File

4.4 Program target device

This tab in the HDL Workflow Advisor is incompatible with The ADI SD card flow. Instead, choose one of the following methods to update the BOOT.BIN file on the SD card.

After the BOOT.BIN file is generated, you have 2 options:

- Copy the BOOT.BIN file on the SD Card directly.
- Send it via network using a terminal (CMD for Windows machine):
    1. Go to the folder where the BOOT.BIN file is:

HighSpeedConverterToolbox/hdl_prj/vivado_ip_prj/boot

    2. Run this command:

scp BOOT.BIN root@<your_board_ip>:/boot

Finally, reboot the board.