

NV HIL Vivado Project

Introduction

Vivado can create such a big and disgusting set of files for a project whose weight can make it clearly inoperative for tasks like version control or for sending it.

Included in this folder are all required files necessary to regenerate the whole Vivado project.

The procedure that must be followed relies in different scripts that will regenerate the whole project. The order in the execution of these scripts is critical because some of them are based in the previous ones.

The following software are involved: Vivado 2017.4 and Vivado HLS 2017.4; other versions of the software may require minor adjustments.

Path assignments

All paths are referred to the main scripts ones. No absolute paths are used during the project.

However, in order to avoid any issue, it is recommended to work in this path: C:\NV_HIL

Vivado HLS

The first step of the whole process is to regenerate all the Vivado HLS IPs. Those IPs are the ones in which the HIL emulator equations are included.

Those IPs are:

Nombre
3ph_modulator_V2
DC_bus_V2
float_convertor_V2
gain_offset_truncation_V2
grid_3ph_inv_V2
grid_pulses_V2
grid_V2
LCL_filter_V2
pmsm_3ph_inv_V2
pmsm_pulses_V2
pmsm_V2
wind_tur_V2

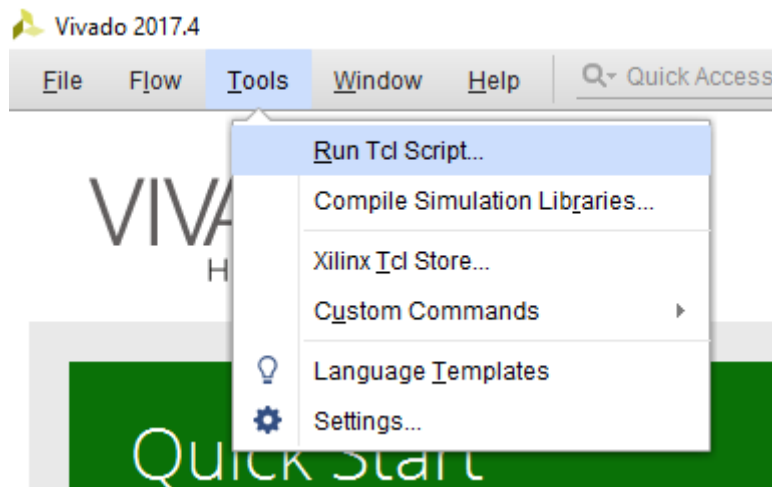
To generate those IPs, the following process must be done with all of them:

1. Open Vivado HLS command prompt
2. Type: `cd C:\NV_HIL\Vivado\HW_repo\HLS\3ph_modulator_V2`
Where "C:\NV_HIL\Vivado\HW_repo\HLS\3ph_modulator_V2" is the path of run.tcl files
3. Type: `Vivado_hls -f run.tcl`
4. Wait for it until it finishes
5. In case it is wanted Vivado GUI to open, type: `Vivado_hls -p 3ph_modulator_V2.prj`
Where 3ph_modulator_V2.prj is the folder where the HLS project has been created.
6. Repeat from 1 to 5 with all the HLS folders

Vivado

To fulfil this step all the Vivado HLS IPs must have been generated.

Open Vivado GUI and click in Tools/Run Tcl Script



Select the file run_prj.tcl and click run.

run_prj.tcl file internally only adjusts the initial path of Vivado and then called the file NV_HIL_prj.tcl which is the one that contains all hardware information of the project.

Wait until the script finishes - It can take up to one hour.

When finished, Vivado will export the bitstream file and will launch SDK.

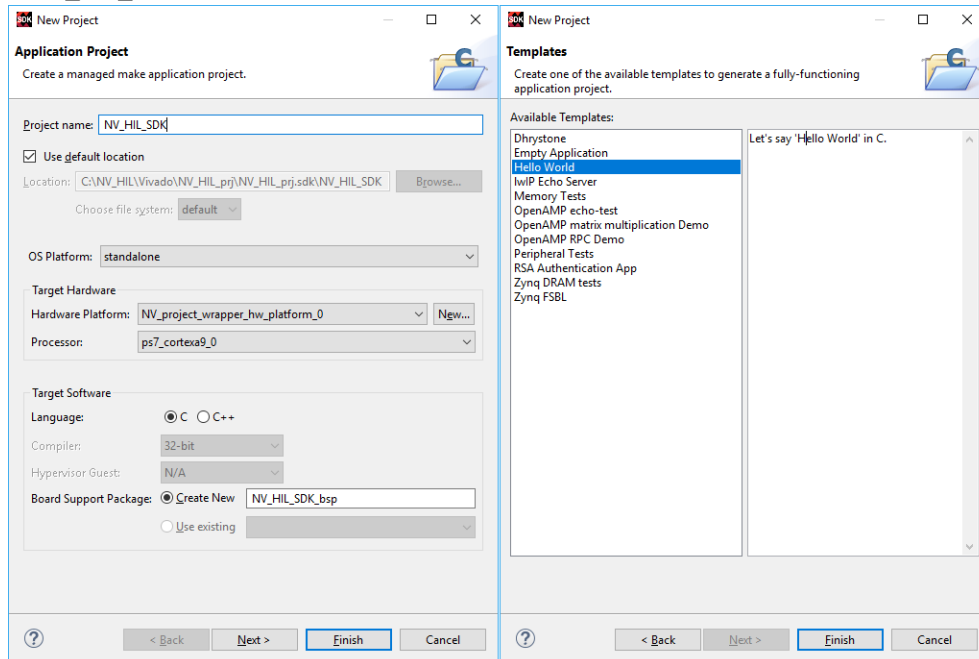
SDK

SDK will configure the ARM microprocessors.

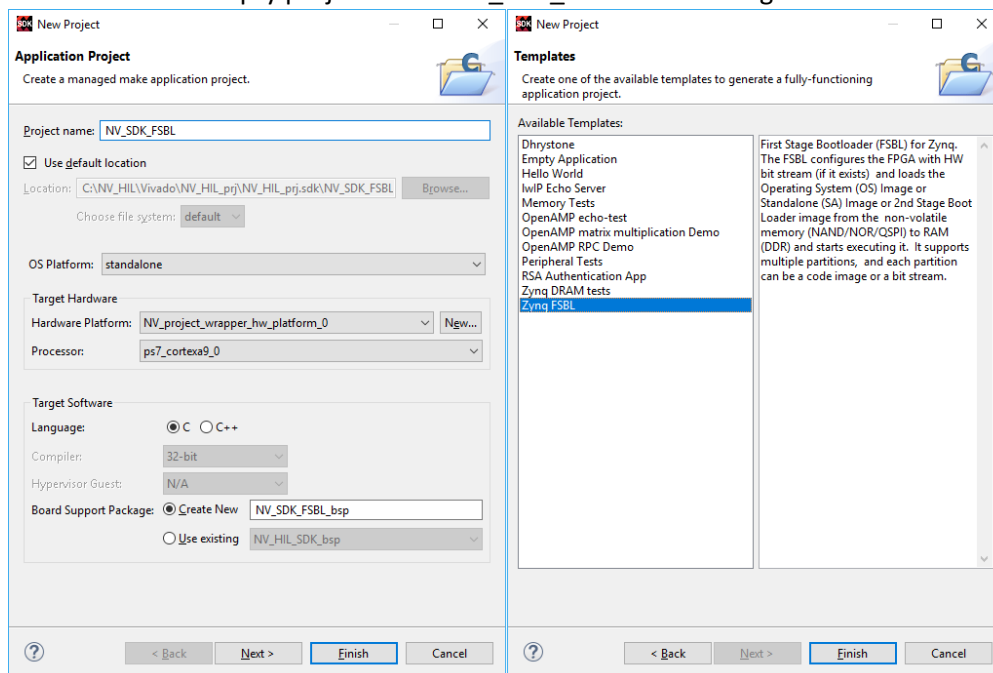
Only one is used and it is in charge of managing the communication with the user desktop console by establishing a serial communication with 1 second of refresh rate.

To configure SDK, the following steps must be accomplished:

1. Create a new empty project by clicking: File/New/Application Project with the name NV_HIL_SDK



2. Copy the contents of provided helloworld.cpp and paste it in SDK new project. Be sure to delete all previous code. This file can be found in SW_repo folder.
3. Create a new empty project called NV_SDK_FSB in order to generate the BOOT.ini file



4. Create the BOOT.ini file by selecting in SDK the folder NV_HIL_SDK and clicking in Xilinx/Create Boot Image menu.

Create Boot Image
Creates Zynq Boot Image in .bin format from given FSBL elf and partition files in specified output folder.

Architecture: **Zynq**

☒ Create new BIF file ☐ Import from existing BIF file

Basic **Security**

Output BIF file path: C:\NV_HIL\Vivado\NV_HIL_prj\NV_HIL_sdk\bootimage\NV_HIL_SDK.bif **Browse...**

UDF data: **Browse...**

☐ Split **Output format:** **BIN**

Output path: C:\NV_HIL\Vivado\NV_HIL_prj\NV_HIL_sdk\bootimage\BOOT.bin **Browse...**

Boot image partitions

File path	Encrypted	Authenticated
(bootloader) C:\NV_HIL\Vivado\NV_HIL_prj\NV_HIL_sdk\N...	none	none
C:\NV_HIL\Vivado\NV_HIL_prj\NV_HIL_sdk\NV_project_wra...	none	none
C:\NV_HIL\Vivado\NV_HIL_prj\NV_HIL_sdk\NV_HIL_SDK\D...	none	none

Add **Delete** **Edit** **Up** **Down**

Preview BIF Changes **Create Image** **Cancel**

5. Go to this path in order to find the BOOT.ini file:
C:\NV_HIL\Vivado\NV_HIL_prj\NV_HIL_prj.sdk\NV_HIL_SDK\bootimage
NOTE: the path may vary.
6. Insert an 8GB microSD card in the computer and format it as FAT32.
7. Paste BOOT.ini file in the microSD card without modifying its name.
8. Remember to configure correctly the microZED board jumpers to boot from microSD card.



Notes

To avoid the user to do the whole regeneration of the project, the following files are provided as they will prove to be useful:

1. BOOT.bin
2. NV_project_wrapper.bit

Both of them are in programming_files folder