



CYPRESS

Method to Instantiate and Use a Core in *LeonardoSpectrum*™

Introduction

This application note is intended to assist people who use cores for Cypress CPLDs and compile their design in *LeonardoSpectrum*™. These cores are distributed using the VIF file format which is generated by *Warp*™. This application note contains a detailed description of how to use cores and associated wrappers in *LeonardoSpectrum*. For more information on how to create a wrapper, please see the application note "Method to Instantiate and Use a Core in *Warp* with Cypress CPLDs".

Preparing *LeonardoSpectrum* for Synthesis

This section is designed to show the user how to create a project in *LeonardoSpectrum* and target a Cypress device.

1. Open *LeonardoSpectrum* and it should appear as in *Figure 1*. The left side of the GUI is where files are added and options are set. The right side of the desktop contains the console and the command window.

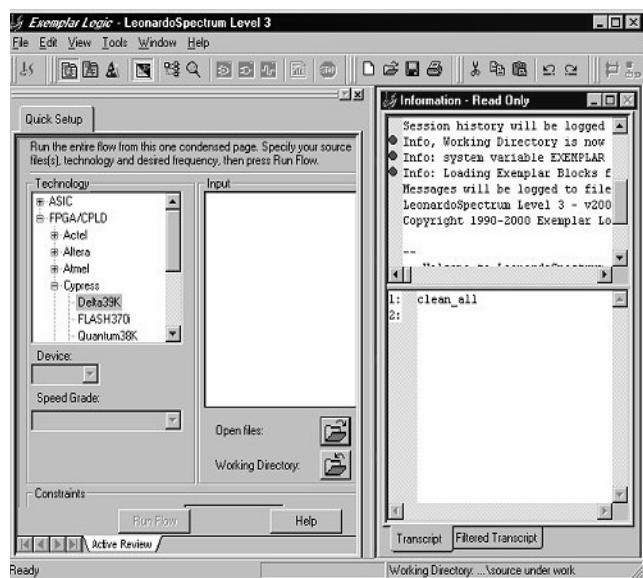


Figure 1. Leonardo Spectrum Desktop

2. If Figure 1 does not appear upon entering *Leonardo Spectrum*, go to "File -> New Project" to create a new project.
3. Save the project by going to "File -> Save Project As.." and select the location where the project will be saved.
4. In the "Technology" window on the left side of Figure 1, select "Cypress" then "Delta 39K" to choose the target device family.
5. Click on the 'Transcript' tab on the bottom right side window of Figure 1.

6. Enter the following line to make the output of Leonardo Spectrum compatible with Warp:

```
set vhdl_write_component_package FALSE
```

Using Cores and VHDL Wrappers in *LeonardoSpectrum*

This section shows the user how to instantiate a core with a VHDL wrapper in *LeonardoSpectrum*.

1. Click on the 'Advanced Flow Tabs' button shown below and *Figure 2* will appear on the left hand side of the desktop.



This is the 'Advanced Flow Tabs'



Figure 2. Advanced Flow Window

2. Click on the '**Output**' tab and choose VHDL as the output file type.
3. In the same window, specify the output file path and file name.
4. Click on the '**Quick Setup**' button shown below.



This is the '**Quick Setup**' button

5. After the "Quick Setup" window opens, click on the '**Open File**' button to add the VHDL wrapper to this project.
6. If more than one file is added to this project, make sure that the top level file is at the bottom of the list.
7. Click on the '**Run Flow**' button to synthesize the design.
8. While *LeonardoSpectrum* is running, the console window will display any errors or warnings.
9. The VHDL wrapper only instantiates the top level component of the core. *Leonardo Spectrum* will pass the component declared in the VHDL wrapper as a black box.
10. The output of this project can be imported into *Warp* and compiled with the VIF files. Please refer to the section "Importing Output of *LeonardoSpectrum* into *Warp*" for more details.

Importing the Output of *LeonardoSpectrum* into *Warp*

1. Create a new Project in *Warp*.
2. Create a new folder in the same directory as the *Warp* project created in Step 1. Name this folder lc<device-name>" where "<devicename>" is the name of the device the VIF files were fitted to. For example, if the VIF files were fit to a 39K100 device, the folder name would be "lc39k100".
3. Copy the downloaded VIF files and the index file to the folder created in Step 3.
4. Add the output of *Leonardo Spectrum* to the project.
5. Compile and fit the design in *Warp*.

Conclusion

The steps described in this application note, allow the user to use Cypress cores together with *LeonardoSpectrum*. This compatibility between Cypress cores and *LeonardoSpectrum* gives users greater flexibility and support.