Method to Instantiate and Use a Core in *Warp Enterprise/Professional™*

Introduction

This application note is intended to assist people who use cores from 3rd party IP vendors in Cypress CPLDs. These cores are distributed using the VIF file format which is generated by *Warp*™. This note contains detailed description of the steps required to instantiate the core contained in the VIF file using *Warp Enterprise/Professional*™ Tool. Please refer to application note "Method to Instantiate and Use a Core in *Warp* with Cypress CPLDs" for additional information.

Creating a Project in *Warp Enterprise/Professional*

1. Upon entering *Warp Enterprise/Professional*, the user is required to create a new design. Go to "File -> New Design..." to obtain *Figure 1*.

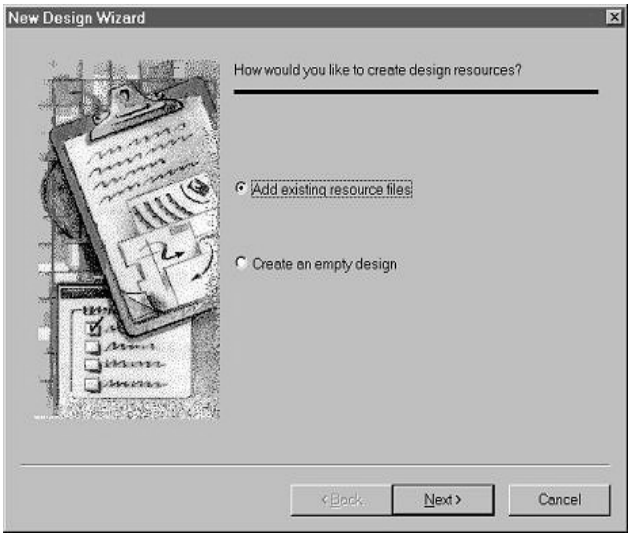

Figure 1. Creating a New Design in *Warp Enterprise/Professional*.

2. The user can choose to add existing resource files or create an empty design. The user should choose to add existing resource files if the downloaded wrapper is used. Follow the instruction in this wizard to create a new design. Choose the default device to be Delta39K.
3. The user can choose the directory where the design is stored. By default, the design will be created in the directory "C:\Cypress\Designs\library name". Where "library name" is the library name created Step 2.
4. The user can choose to create a new library within a design. Go to "Design -> Create Library...". Follow the instructions on the create library wizard to create the library. Choose the path of the library. Usually, it should be the same path as the design mentioned in Step 3.

Preparing VIF Files for Use in *Warp Enterprise/Professional*

When a user receives a core package from a 3rd party IP Vendor, the Cores are in the form of VIF files. VIF files are used by *Warp* to describe functional blocks e.g cores. The core package consists of core VIF files and VHDL or Verilog wrappers.

1. Start *Warp Enterprise/Professional* and select "View -> Flow". *Figure 2* will appear and the user can click on the different component of the design flow to accomplish the task described in the figure.

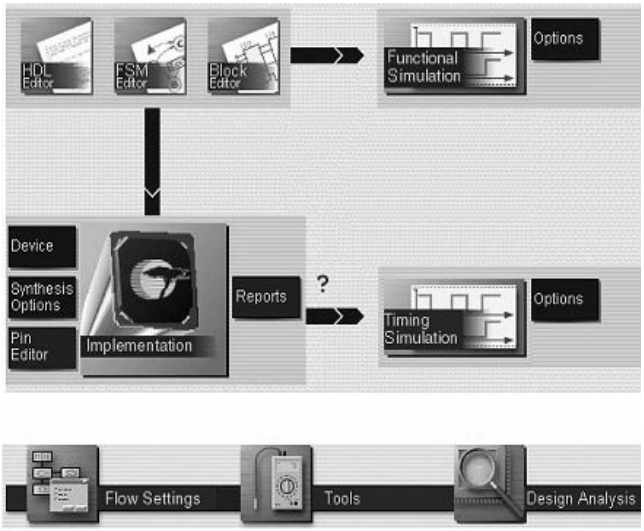

Figure 2. Design Flow Diagram in *Warp Enterprise/Professional*.

2. In *Figure 2*, click on the '**Tools**' picture, then the '**Synthesis Library Manager**' picture shown by the arrow in *Figure 3*. A library window will automatically appear and *Warp Enterprise/Professional* will automatically create a *Warp* library. The content of the *Warp* library is stored in a synthesis folder under the same directory as the design created in section " Creating a project in Warp Enterprise/Professional".

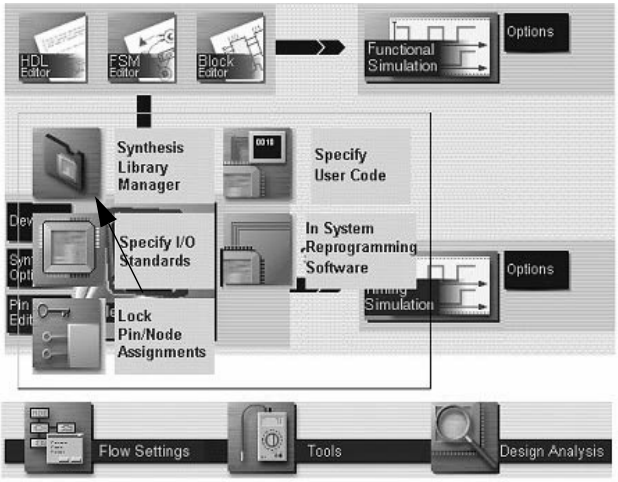

Figure 3. *Warp* Synthesis Library

3. Click cancel to close the library window.
4. In the same synthesis folder created in step 2, copy the downloaded VIF files and the index file to a new folder named "lc<devicename>" where "<devicename>" is the name of the device. For example, if the VIF files were fit to a c39k100 device, the folder name would be "lc39k100". The index file is automatically generated by *Warp* upon *Warp* project compilation.

How to Compile and Fit a Design Using Downloaded Wrapper and VIF Files in *Warp Enterprise/Professional*

1. Add the wrapper VHDL or Verilog file to the Warp project. Instantiate the wrapper in the user's own design. The user can also compile the wrapper in *Warp* by itself. The process for both is the same.
2. After all the files are present in the library, click the '**Synthesis Option**' button in *Figure 2* to make sure that the synthesis options are set correctly. It is suggested that the user disables the "logic to memory mapping".
3. Specify the target device by clicking on the '**Device**' picture in Figure 1 to select a device.
4. Click on the '**Implementation**' picture in *Figure 2* and another window will come up displaying all the files in the library. Select the files to be compiled and synthesized in *Warp*. Hit '**OK**' and this will link to *Warp* and synthesize the design in the library using *Warp*.
5. The user can hit the '**Report**' button to view the report file.

How to Use the Core for Prefit Simulation in *Warp Enterprise/Professional*.

The user can do a prefit simulation with real timing values for cores in *Warp Enterprise/Professional*. This means that the user's design is functionally simulated and only the post-fit model of the core contains propagation delays. The user's design is not fit to a Cypress CPLD.

1. Create a new project as directed in section "Creating a Project in *Warp Enterprise/Professional*".
2. Compile and fit the wrapper of the core by following the entire procedure outlined in section "How to compile and fit a Design Using Downloaded Wrapper and VIF files in *Warp Enterprise/Professional*".
3. Go to the directory ...\synthesis\vhd to obtain the postfit netlist of Step 2 and import it into a new project.
4. Instantiate this postfit netlist in the user's design. Make sure that the component and IO declaration of the postfit netlist matches the instantiation in the user's design.
5. If the original wrapper has generics, the postfit netlist will not contain any generics. The user must compile and fit the wrapper with the generic values wanted for the design. This involves changing the default values of the generics in the original wrapper and compiling the wrapper with the change.
6. Go to "Design -> Compile All" to compile all the files in the project.
7. Write an appropriate testbench and instantiate the design containing the core in the testbench.
8. Go to "Simulation -> Initialize Simulation". This will allow the user to pick the top-level file to be simulated. This top-level file is the testbench file.

Note: Even though this is a prefit simulation, there are time delays in the core as it is a post-fit netlist.

Creating a Block Diagram in *Warp Enterprise/Professional*

A block diagram can be created for any VHDL component. This application note will use the VHDL wrapper as the source of the block diagram to be created.

1. Create a new design and follow the instructions given in the section "How to Use the Core for Prefit Simulation in *Warp Enterprise/Professional*" to prepare the VIF files in this new design.
2. Import the VHDL wrapper file in this new design.
3. Highlight this file in the Design Browser and go to "Design -> Compile" to compile this file.
4. When the design has successfully compiled, a check mark should appear beside the wrapper file in the design browser.
5. Click on the drop-down arrow of the '**New**' button and choose '**Block Diagram**' as shown in *Figure 4*, by the arrow. This will create a new block diagram editor without going through the wizard.

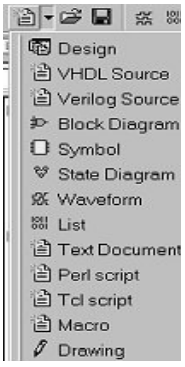

Figure 4. New Button Drop-down display

6. Once a new Block editor file comes up, click on the '**Show Symbols Toolbox**' button as shown below. This will bring up a list of symbols the user can import into the block editor.

This is the '**Show Symbols Toolbox**' button.

7. In *Figure 5*, pick the entity of the Wrapper file. Click on the yellow rectangular component as pointed to by the arrow and drag it to the Block Editor file. A generic rectangular block is instantly generated with the correct port names.

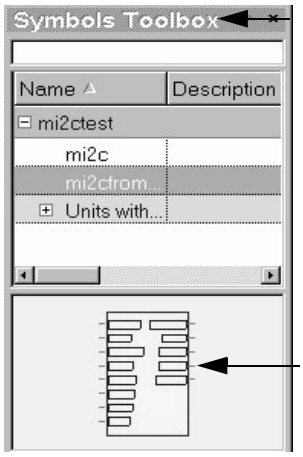

Figure 5. Symbols Toolbox of the Block Editor.

8. The user can then use the '**Block Editor**' feature of *Warp Enterprise/Professional* shown in *Figure 2* to do his/her design with the core wrappers.

Testing The Block Diagram

After creating the block diagram based on the VHDL file described in the previous section, the user may wish to compile the block diagram in Warp to obtain timing values and to test the functionality of the design. In the example given in this application note, the block diagram is for a core.

1. Create a new Block Editor File by following Step 5 in the section "Creating a Block Diagram in *Warp Enterprise/Professional*" and instantiate the block by following Steps 6 and 7 in the same section.
2. Go to the '**terminal**' button as shown below. Click on the drop-down arrow to obtain different input and output terminals.

This button varies according to the last terminal selected.

3. Attach the appropriate input and output terminals to the block diagram. The user should use the bus input and output terminals for any std_logic_vector. An example of the final block diagram is shown in *Figure 6*.

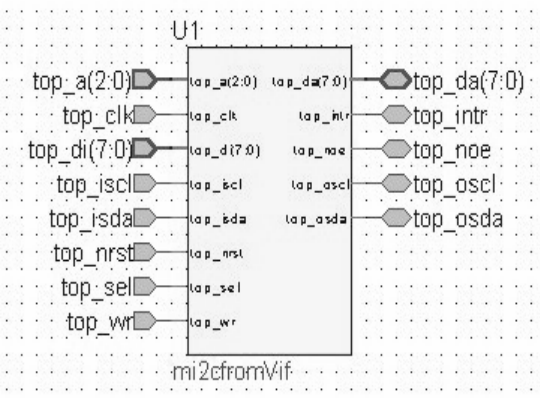

Figure 6. Block Diagram of the Wrapper mi2c.

4. After the user is done with their design and wishes to create a file in either VHDL, Verilog or EDIF based on this block design, go to "Diagram -> Set Target HDL.." and click on the appropriate language to generate.
1. Save the block design and go to "Diagram -> Generate HDL Code" to generate the VHDL code.

Conclusion

This document illustrates the steps required to use *Warp Enterprise/Professional* with cores. Having pre-verified and optimized solutions for functional blocks enables designers to concentrate on developing the custom logic, resulting in a significant reduction in design cycle time.