Reference Project Library User Manual

Revision: 3/21/05

www.digilentinc.com
246 East Main | Pullman, WA 99163
(509) 334 6306 Voice and Fax

The Reference Project Library

This document explains how to read, modify, compile, and use the projects in the *Digilent Reference Project Library*.

Digilent has created a library of reference projects for use with Digilent boards. They implement typical applications that demonstrate the boards' features. The reference projects also demonstrate and test the behavior of some general interest VHDL and schematic modules (components) found in the *Digilent Component Library*.

Digilent continually adds new reference projects and updates them for additional board configurations.

These following projects are currently available.

C1MemCfg
A configuration project for the C1 Memory Module (512 KB RAM + 512 KB Flash). It also works with the C0 Memory Module (128 KB RAM + 512 KB Flash).

C1MemCfgX2
A configuration project for two C1 Memory Modules (512 KB RAM + 512 KB Flash) attached to the same motherboard. It also works with C0 Memory Modules (128 KB RAM + 512 KB Flash) or a C0/C1 combination.

C2MemCfg
A configuration project for the C2 Memory Module (2 X 512 KB RAM, seen as a contiguous 1 MB RAM space).

C3MemCfg
A configuration project for the C3 Memory Module (1 MB Flash).

The Project Files

Each reference project can be downloaded from www.digilentinc.com as an archive with files in a hierarchical folder structure. The top-level folder is labeled with the project's name. It includes a *Project User Manual,* a *Project Design Document,* and a subfolder list similar to the one shown here:

Project_Name
- Bit_files
- Board_Specific_Source_Files
- xxx_Project

The **xxx_Project** folder includes a ready-to-compile project for the "xxx" board. If additional peripheral boards are required, the subfolder name would refer to the board names and the associated connectors. For example, the **USB-A1_C1Mem-A2_DIO4-C1c2** folder shows that the USB module needs to be attached to the system board's A1 connector, the C1 Memory Module needs to be attached to connector A2, and a DIO4 peripheral board needs to be attached to connectors C1 and C2.

The project contains all the source files (.vhd, .sch. .sym, .ucf) and the project files (*project_name.npl*), as well as the initial structure of the standard **__projnav** folder. The project is set for the selected main/peripheral board configuration, but has never been compiled. To save space, the intermediate files, generated by the WebPack software, are not included. To compile the project, you only need to select the top-level file and launch the appropriate process (e.g., Generate Programming File).

The **Board_Specific_Source_Files** folder contains source files specific to board combinations used other than the one in the **xxx_Project**. Two hierarchical levels of folders name the system board, the peripheral boards, and associated system board connectors, respectively.

The folder includes .ucf files, but sometimes includes other source files to help you adapt the **xxx_Project** to your own configuration.
To adapt the project:

copy the files in the specific subfolder of **Board_Specific_Source_Files** and paste them to the **xxx_Project**, (replacing the existing same-name files).
set the Project Properties, according to the FPGA equipped on the system board.
change the **xxx_Project** folder name (or make a copy of it) to reflect the file changes.
compile the modified project to obtain the configuration files.

The **Bit_files** folder contains a collection of compiled .bit files implementing the project, each specific to the board combination used. Two hierarchical levels of folders name the system board, the peripheral boards, and associated system board connectors, respectively.

Each Reference Project is documented with a *Project User Manual*. The user manual explains the Project behavior and usage. Complementary software programs are indicated, when necessary.

The components used are documented in a comment block (the header of the .vhd file) or in a separate document for .sch components. More information about individual components can be found in the *Digilent Component Library*.

Working With the Reference Projects

Before attempting to use the project as described below, please read the *Project User Manual*. Check if the project behavior matches your needs and expectations. If not, you will need to modify the project (see paragraph 6), but first check that all the required hardware and software resources are available.

A project can be used in several different ways:

1. Use the pre-compiled .bit file.
   1.1. Find the appropriate .bit file. Browse the **Bit_files** folder structure for the desired system board and peripheral board combination. The first hierarchical level in the **Bit_files** folder structure encodes the system board name. If peripheral boards are required, the next hierarchical level folder shows the board names and associated system board connectors. If the desired board combination is not available, you will need to transport the project to the desired board combination (see paragraph 3).
   1.2. Download the project onto your system board and use it.

2. Compile the project.
   2.1. Check if the ready-to-compile project is prepared for the desired board combination. If not, you will need to transport the project to the desired board combination (see paragraph 3).
   2.2. Check if the ready-to-compile project is prepared for the desired HDL compiler. (The comment block at the beginning of each .vhd file specifies the software used to test the component, typically a version of Xilinx WebPack.) If not, you will need to transport the project to the desired HDL compiler (see paragraph 4) or build a new project from the uncompiled components (see paragraph 5).
   2.3. Open the project with the desired HDL compiler (Project Manager).
   2.4. Select the top level hierarchical file. Usually, this file is named *Main.vhd* or *Main.sch* and is shown at the top level in the Project Manager *Sources in Project* window. However, there are projects offering multiple compilation variants. For more details, see the *Project User Manual*.
   2.5. Compile the project to obtain the configuration file (.bit).
   2.6. Download the project onto your system board and use it.

3. Transport the project to the desired board combination.
   3.1. If your particular board combination is not the one used in the **xxx_Project**, find it in the **Board_Specific_Source_Files**. Copy all the source files found in the

specific folder (.ucf and possibly other source files) to the project folder, replacing the existing ones. Continue with step 2.2, above. After opening the project, set the Project Properties according to the FPGA chip and package on your system board.

3.2. If your particular board configuration is not provided (neither in the **xxx_Project**, nor in the **Board_Specific_Source_Files**) check if the configuration supports the project:
- The required IO devices are available.
- The software is available.
- The number of external connectors is appropriate.

If so, adjust the project to fit your board configuration (basically the .ucf file(s) and Project Properties) and recompile the project. Additional .ucf patterns can be found in the *Digilent Component Library* in a folder structure using the component name and the host Digilent board. In some cases, other source files need to be slightly modified to fit the new configuration. If the project is too large to fit in the FPGA on your motherboard, try modifying the project by removing unessential components and/or features (see paragraph 6).

4. Transport the project to the desired HDL compiler.
The comment block at the beginning of each .vhd file specifies the software used to test the component, typically a version of Xilinx WebPack. If your compiler is a later version of WebPack, the Project Manager will warn you that the original project was compiled in a previous version. Follow the on-screen instructions to upgrade the project for the new WebPack version. Usually the process does not generate problems. However, we did notice some problems when compiling the same source files with successive WebPack variants. To avoid such problems, make sure you understand the different features of the new software variant. Any problems encountered while the project was tested are noted in comments close to the implied code lines.

5. Build a project starting from uncompiled components.
   5.1. Create a folder for the project (make sure to obey the general rules for a Project Manager path, e.g., no spaces in the folder names, etc.)
   5.2. Copy the all the source files (the .vhs, .sch, and .sym files found in the **xxx_Project** folder) to the project folder.
   5.3. Copy the appropriate .ucf file(s), depending on the board combination, to the project folder.
   5.4. If the .ucf file is not available for your board combination, edit it (.ucf patterns are available in the *Digilent Component Library* for each component for several Digilent board combinations).
   5.5. Launch the Project Navigator.
   5.6. Set the project name.
   5.7. Browse to set the project location to the folder created above.
   5.8. Set the appropriate Top-Level Module type (for Digilent Reference Projects Library, see the project description document).
   5.9. Set the device, corresponding to the used motherboard (Project Properties).
   5.10. Add all the sources shown by the wizard (Add Source, ctrl A, Open).
   5.11. Set the .vhd files type (usually VHDL Design File).

5.12. Assign the .ucf file to the appropriate design file (usually the top level file, having the same name) (for Digilent Reference Projects Library, see the project description document).
5.13. Select the highest level design file in the hierarchical structure (for some projects multiple options are available; see the project description document).
5.14. For JTAG downloaded bit file:
  5.14.1. Click right on Generate Programming File in the Processes Window.
  5.14.2. Choose Properties.
  5.14.3. Choose the Startup Options tab.
  5.14.4. Set FPGA Startup Clock property to JTAG Clock value.
  5.14.5. Click OK.
5.15. Compile the project.

6. Modify the project.
Keep in mind that you are on your own! While Digilent provides design documents and open sources for the projects, we can not provide support for user modifications. When modifying the source files, make sure you understand the project behavior, check the behavior after each step, and keep track of changes.