Cerebrum GUI User Guide
Version 3.00.a

PSU Engineering (Cerebrum) Team

Created: 23 Oct 2010

Updated: 26 Jun 2011

CHANGE LOG

Version	Date	Name	Description
1.00.a	23 Oct 2010	M. Cotter	Created the document
2.00.a	23 Oct 2010	M. Cotter	Added documentation for the GUI and design flow
2.00.a	18 Nov 2010	M. Cotter	Corrected version number and table of contents
3.00.a	26 June 2011	M. Cotter	Added information on dialogs for synthesis and XPS builder options.

Table of Contents

1 Setting up the Cerebrum Graphical User Interface

This section provides information on how to setup and configure the Cerebrum Graphical User Interface and design Framework for use.

1.1 Package Installation

Download the Cerebrum Package from https://. Unpack the archive to the location where you want to install the tool. For example, if you want to install the tool into "C:\Cerebrum" place the archive in that folder and extract it "here".

1.2 Configuration of the Tool Environment

Open the 'InstallPath.reg' file and edit the "InstallPath" registry value to point to the directory where you have unpacked the Cerebrum archive. This directory would be the same as the directory where the 'InstallPath.reg' file is located. Note, that in this file all backslashes must be escaped as double-backslashes. For example, "C:\Cerebrum" should be entered as "C:\\Cerebrum". After saving this file, run it to import this information into the Windows registry.

The next step is to edit the default project paths file used by Cerebrum. This file is located in "defaults\install_paths.xml". Set the "CerebrumRoot" path to the same directory as the path entered in 'InstallPath.reg'. In this file, do not use double-backslashes. The remaining paths in this file, are pre-defined relative to the CerebrumRoot path, and are included as part of the install package.

Once the registry file has been updated and merged, and the default paths have been updated, the tool is ready to use and may be started by running the "CerebrumFrontEndGUI.exe" program in the 'bin' subdirectory of the install location.

2 Cerebrum Graphical User Interface

This section provides an introduction to the Cerebrum Graphical User Interface and design Framework.

2.1 The Primary Design Workspace

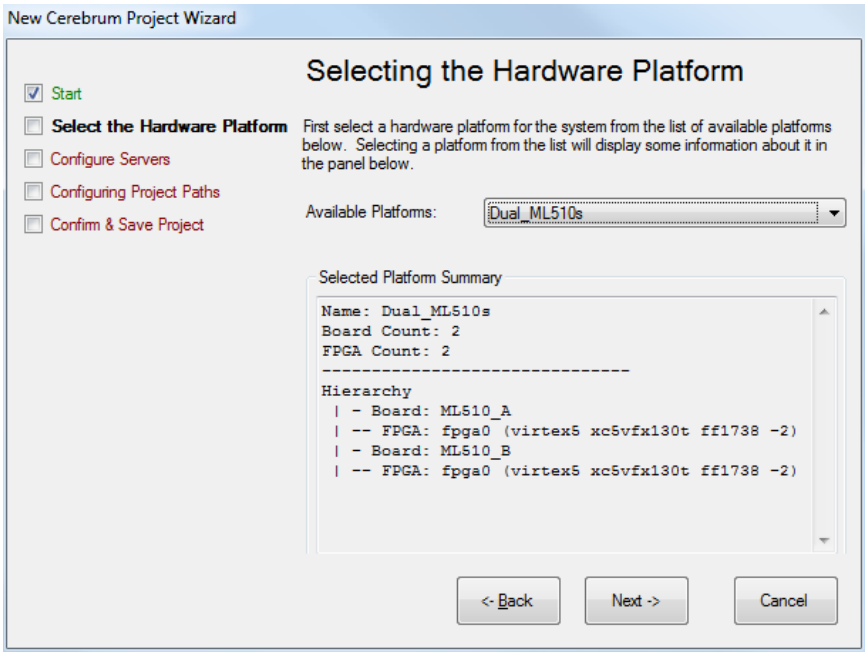
Upon starting the tool, the primary design workspace is displayed to the user. At this point, since no project is loaded, very few options are available: Create a New Project and Open an Existing Project.

2.2 The New Project Wizard

The new project wizard facilitates easy creation of Cerebrum projects and the necessary directory and file structure of the project. The project wizard guides the user through three essential steps that are necessary to define a basic Cerebrum project.

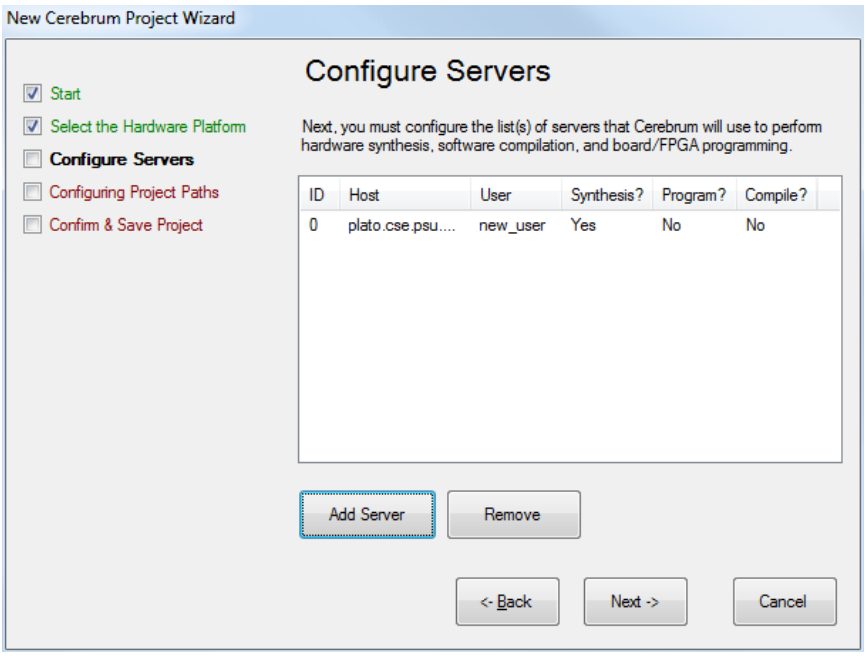
2.2.1 Platform Selection

The first stage of the new project wizard is the selection of the hardware platform on which the design will be created. This dialog provides a drop-down list of the available platforms and a brief description of the Boards and FPGAs defined within the platform. Once the appropriate platform has been selected, the user can proceed to the server definition stage of the new project wizard.



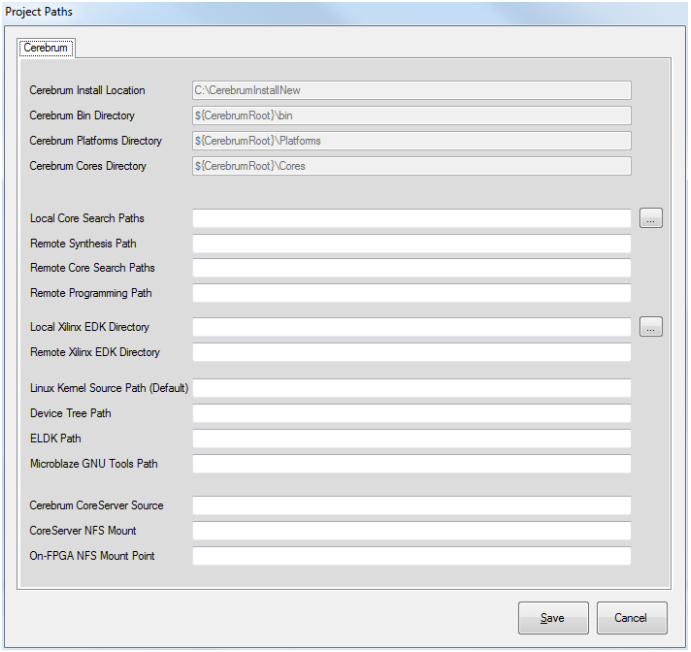
2.2.2 Server Management

The second stage of the wizard allows the user to define the synthesis, programming, and compilation servers that will be used to produce the final design on hardware. On this screen, the user can add to, remove from and modify the list of servers available, and their functionality. Each server must be assigned to at least one role (Synthesis, Programming, or Compilation). The address and login id of each server can be modified at this point as well. The list of servers may be updated and modified once the wizard has completed. If the available servers are not known at this point, the list may be left empty, but none of synthesis, programming or compilation can be completed successfully until a server has been defined and assigned to the corresponding role at a later time. Once the server list has been populated the user may proceed to defining the project paths.



2.2.3 Project Paths

The third stage of the wizard allows the user to define the paths to essential project directories, both locally and remote that are used in locating Cerebrum cores, building and synthesizing the XPS projects, programming the FPGAs and configuring and compiling software. These paths can be updated and modified later as necessary. This stage of the wizard also provides the user with the option of importing the paths from an existing project, if the user wants to use the same locations as a previous project. Once this step has been completed, the user may move on to saving the project.



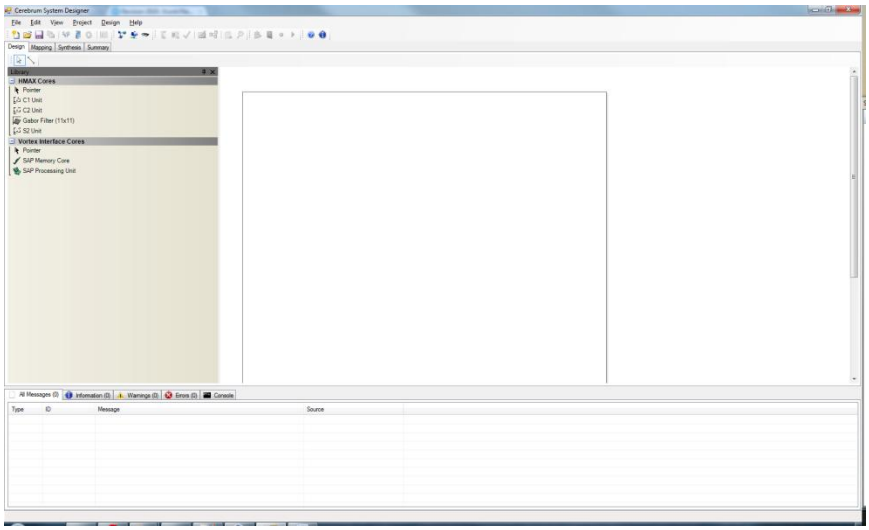
2.2.4 Saving the Project

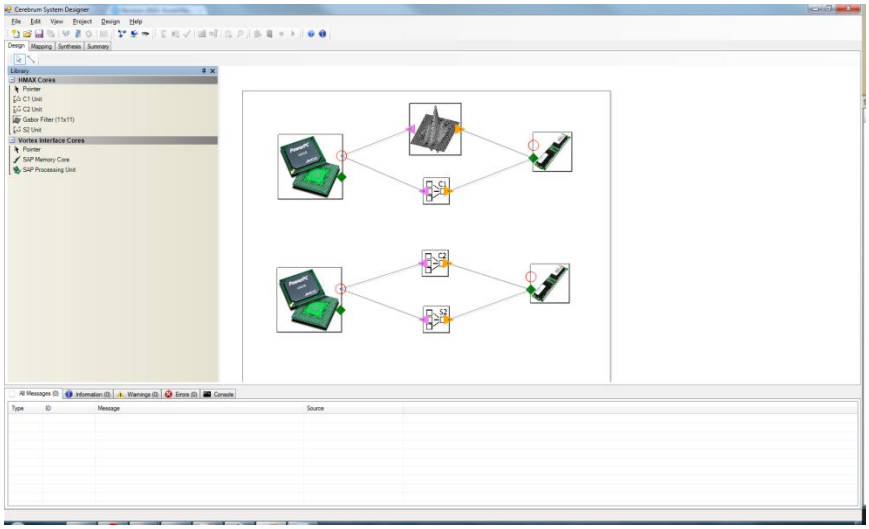
The final stage of the project wizard is selecting the location in which the project file(s) are to be saved. Clicking the button with the ellipsis will provide the user with a dialog that allows for selection/creation of a folder in which to save the project. Once the folder has been selected, clicking the 'Finish' button will create the project structure and open the newly created project.

2.3 The Primary Design Workspace (Open Project)

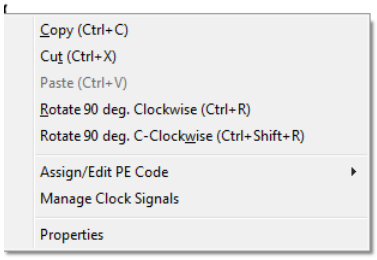
Once a project has been created and opened, the design workspace now displays the design canvas as well as the Design library toolbox. At this point, the user may drag design cores from the library toolbox and drop them on the design canvas. The component cores available in the library toolbox are parsed from the default Cerebrum core repository and the ProjectCores project path.

As components are dropped on the canvas, the connector tool on the design toolbar allows the user to draw connections between the components. Connections should be drawn from the source/initiator component to the target. The connection will automatically attach to the nearest compatible port available on the source and sink cores. This magnetic attachment of the connectors to the ports means that each connector does not have to be on the port to which it can be connected, but rather simply hit the component that it's to be connected to.





2.3.1 Design Actions



2.3.1.1 Copy/Cut/Paste

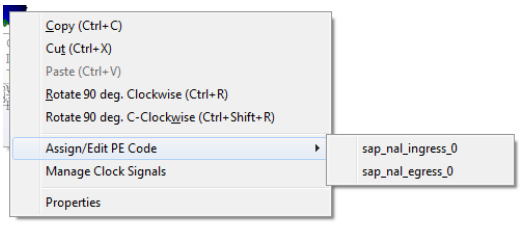
These functions allow for copying, cutting, and pasting components in the design. Any pasted components retain the properties of the component from which they were copied, except for their internal instance name, which is automatically generated.

2.3.1.2 Rotate Clockwise/C-Clockwise

These functions allow the component to be rotated within the design. This does not impact the design in any way, but may make the connections easier to follow and/or see in a complicated system.

2.3.1.3 Assign/Edit PE Code

For components that contain Processing Elements (PEs), this option will be available to assign and/or edit C-codelets attached to the PEs within the component.



2.3.1.4 Manage Clock Signals

This option allows the user to manually assign any clock inputs to the component to the output clocks of other components in the design, if any are available. By default, these signals will be sourced by any available clock net in the design, or if no compatible clocks are available, by an automatically inferred clock generator core.

2.3.1.5 Properties

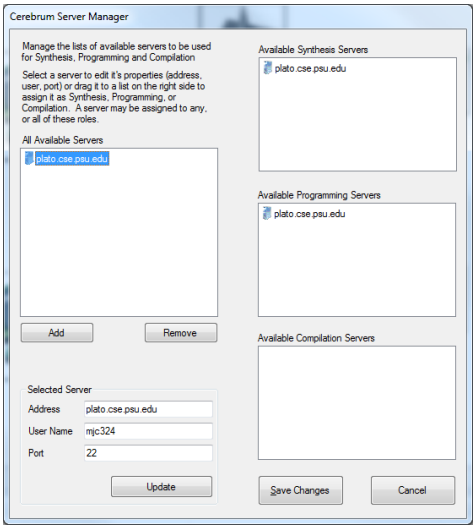
The properties option displays the component properties dialog, which allows the user to manually set any values on properties exposed by the component package.

2.4 Project Paths Management

The project paths can be configured while the design is loaded. Access to the dialog can be found under the Project->Path Settings... menu as well as on the Project Toolbar. The dialog is the same as the dialog displayed during the new project wizard.

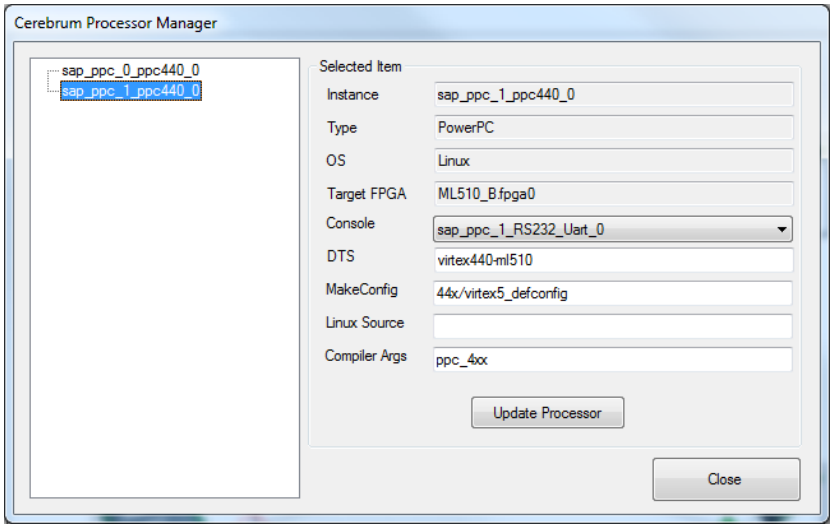
2.5 Project Servers Management

The project servers can be configured while the design is loaded. Access to the dialog can be found under the Project->Edit Server Lists... menu as well as on the Project Toolbar. This dialog is not as compact as the one displayed during the project wizard. Assignment of servers is done by dragging the server from the global servers list to the assignment-specific server lists. Selecting a server on any of these lists, allows the user to edit the address, port and login id of the server. Adding a new server adds it to the global list, and it must be assigned a role before it will be saved. Dragging a server from any assigned server list to the global list will remove its assignment from that role. Removing the selected server will remove it from all lists.



2.6 System Processors Management

As cores are added to the design, if any are found to contain processors (such as Microblaze or PowerPC), they require additional configuration in order to compile the embedded Linux for them to run on the hardware. Access to the configuration options for these processors can be found in the Design menu or on the Design toolbar.

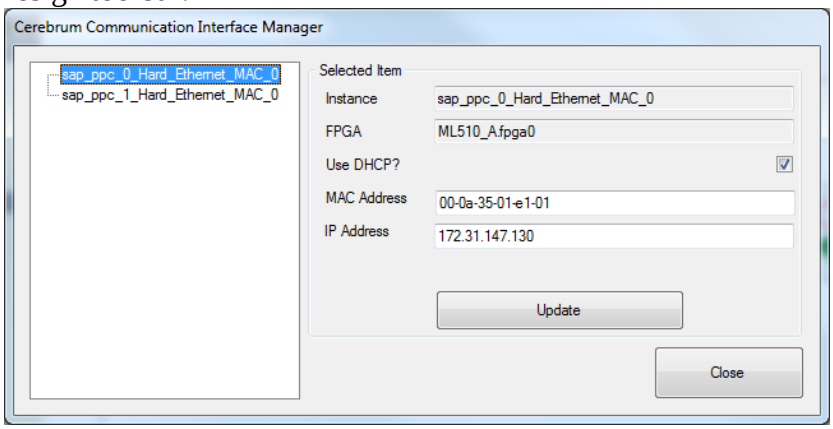


The type of the processor is automatically identified when the processor is detected, and the OS is currently fixed to be Linux. The target FPGA is read-only and is automatically set when the component which contains the processor is mapped to an FPGA (See Component Mapping). The list of available console devices is populated based on the compatible cores contained within the same component as the processor. The DTS file parameter specifies the name of the Device Tree Specification that is to be generated for the Linux kernel compilation. The MakeConfig parameter specifies the default configuration settings to be used for preparing the Linux kernel for compilation. The Compiler Args specifies the arguments to be passed to the ELDK cross compilation initialization script. Finally, the Linux Source parameter allows the user to specify

a custom Linux source tree to be used, overriding the default global source as specified in the project paths file.

2.7 Communication Interface Management

As cores are added to the design, if any are found to contain an Ethernet device (TEMAC), they require additional configuration in order to compile the embedded Linux for them to run on the hardware. Access to the configuration options for these processors can be found in the Design menu or on the Design toolbar.



For each device, the FPGA field is read-only and is automatically set when the component which contains the processor is mapped to an FPGA (See Component Mapping). The Use DHCP check box determines whether the device will be configured to request an IP address when booting Linux. The MAC address field indicates the MAC that is to be assigned to the Ethernet device in Linux. Finally, the IP address is used to indicate the static IP address that will be assigned to the device in Linux if DHCP is not enabled. If DHCP is enabled, this value should be set to the IP address that would be assigned to the specified MAC address.

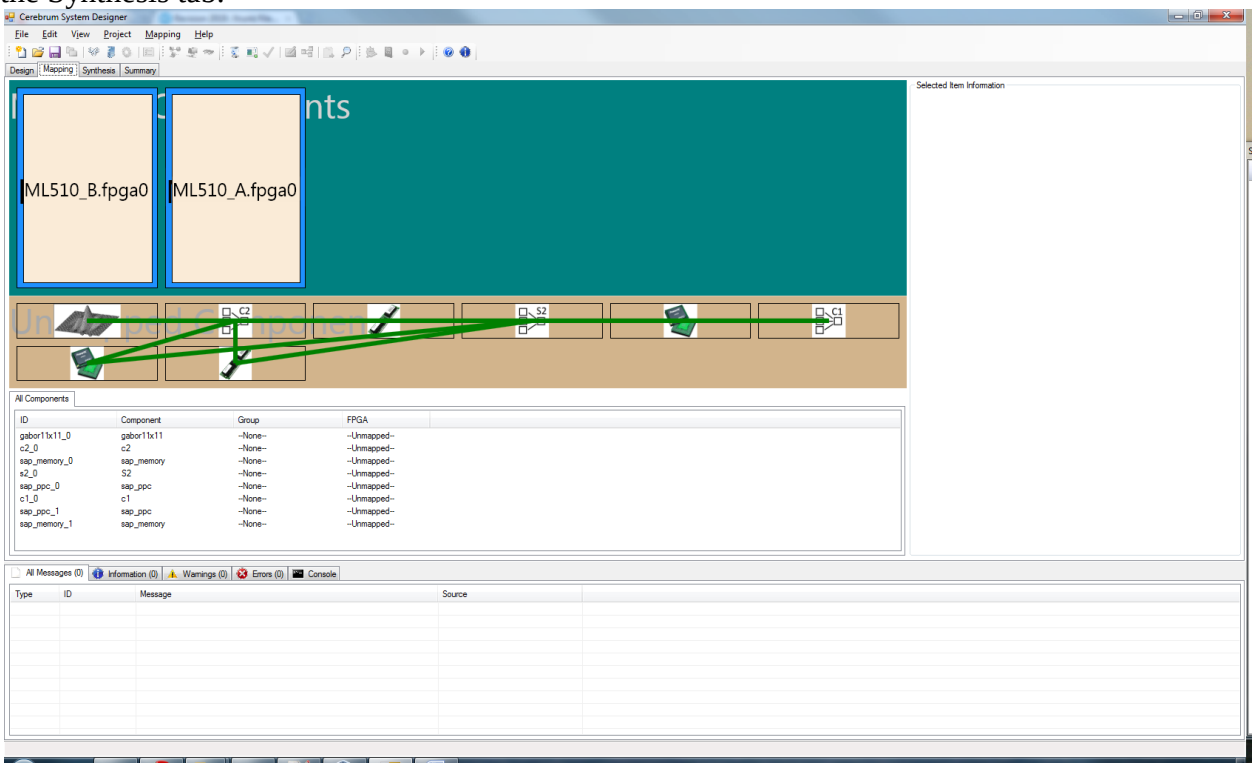
2.8 Component Mapping

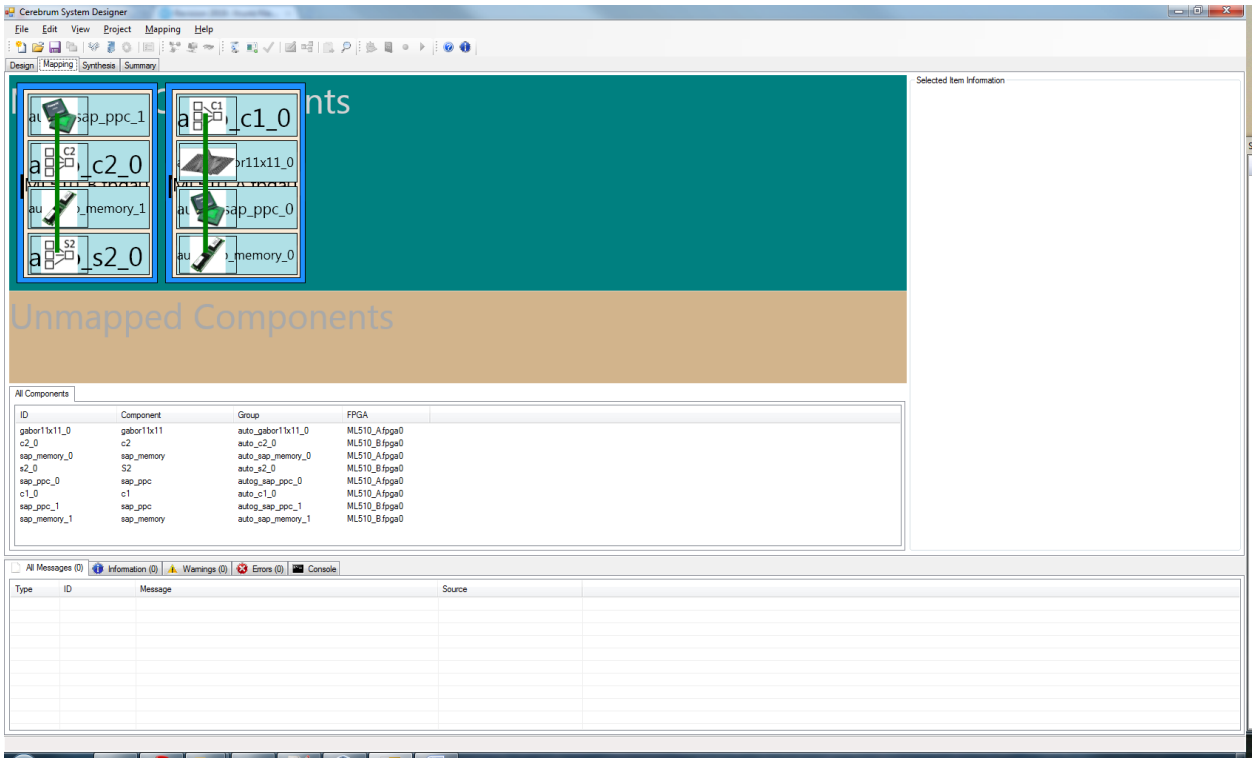
Selecting the Mapping tab will activate the Mapping interface that allows for manual or automatic mapping of components to FPGAs defined in the platform. The mapping interface provides two modes of operation. The first, and default, is Manual Mapping mode. This mode allows the user to manually map components to FPGAs. The second mode, Query mode, allows the user to select FPGAs and components to view their resource requirements and availability.

Currently, in platforms that contain FPGAs that are not connected via some path to all other FPGAs in the platform, may not be able to automatically map all components of a multi-part disconnected system. In these situations, it may be necessary for the user to manually map parts of each disconnected system to a disconnected subset of FPGAs within the Platform.

As an example, in the design shown in section 2.3, two disconnected subsystems are defined. As the platform has 2 disconnected FPGAs, the user will have to manually map part of each system (the PPC units, for example) to each FPGA. From this point, the mapping algorithm will be able to map the remaining pieces to the correct FPGAs, assuming sufficient resources are available on them.

Once any user-directed mappings have been made, the mapping needs to be completed and confirmed. Confirmation of the mapping assignment will update and generate the required files for post-mapping hardware synthesis and software compilation. The component mapping must be completed successfully prior to beginning hardware synthesis and software compilation. Once mapping has been completed, the design may move to the Synthesis phase by activating the Synthesis tab.



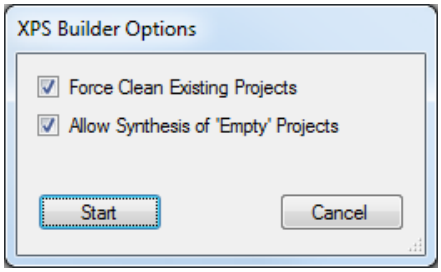


2.9 XPS Project Building

To begin post-mapping synthesis, the user needs to activate the Synthesis tab. The first step in this process is to build the XPS projects. This is done from the GUI by clicking the Build XPS Projects button on the Synthesis toolbar or selecting it from the Synthesis menu.

When activated, a small dialog with a few options specific to the XPS Builder is displayed prior to starting the tool. Also, the tool will request a password via a popup dialog for the login and server it requires to complete the process.

2.9.1 XPS Builder Options



2.9.1.1 Force Clean Existing Projects

This option forces the XPS Builder to remove all files related to these projects when transferring the files to the synthesis server.

2.9.1.2 Allow Synthesis of 'Empty' Projects

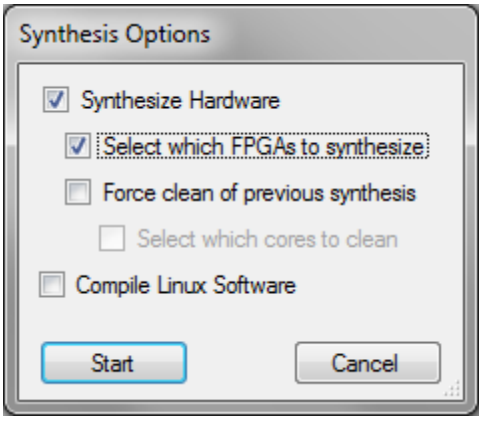
This option allows the XPS Builder to create and synthesize projects for FPGAs which do not have components mapped to them.

2.10 Synthesis

The next step is to being project synthesis and compilation. This is done from the GUI by clicking the Synthesize XPS Projects button on the Synthesis toolbar or selecting it from the Synthesis menu.

When activated, a small dialog with a few options specific to the System Synthesizer is displayed prior to starting the tool. Also, the tool will request a password via a popup dialog for the login and server it requires to complete the process. All FPGAs will be launched for synthesis in parallel on the available synthesis server(s).

2.10.1 Synthesis Options



2.10.1.1 Synthesize Hardware

When enabled, the synthesis tools will generate a new hardware bitstream using the Xilinx synthesis tools, including the compilation and merging of PE codelets into the new bitstream. When disabled, only PE codelets will be compiled and merged into a previously created bitstream. If such a bitstream does not exist (a previous synthesis run was not successful since the last clean), this will fail.

2.10.1.2 Select which FPGAs to synthesize

This option allows the user to select specific FPGAs within the platform to be synthesized, if only a specific subset is required.

2.10.1.3 Force Clean of Previous Synthesis

This option forces the synthesis tools to clean any previously synthesized or cached files from previous synthesis runs prior to beginning the new synthesis run. If the 'Select which cores to clean' option is not enabled, ALL previous synthesis files are cleaned.

2.10.1.4 Select which cores to clean

Available only when 'Force clean of previous synthesis' is enabled, this option allows the user to select specific cores mapped to each FPGA to be cleaned, rather than purging all of the previous files.

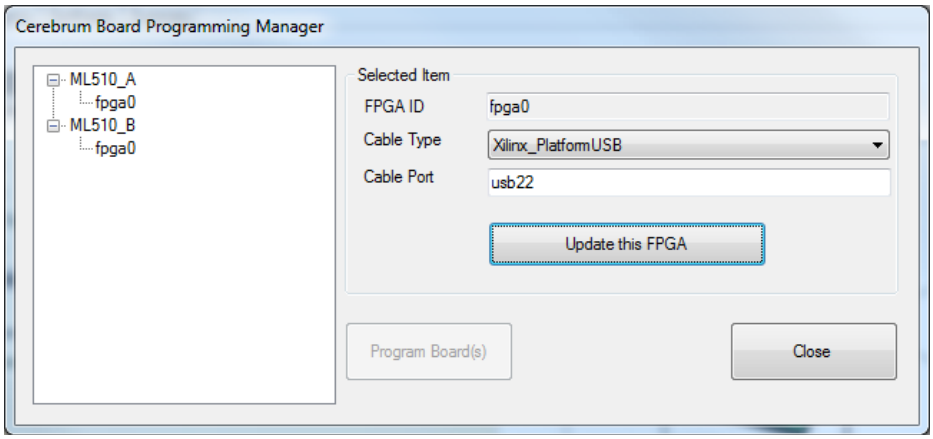
2.10.1.5 Compile Linux Software

This option causes the tools to compile an ELF file to run an embedded Linux operating system on any available PowerPC or Microblaze processors found and configured on the system.

2.11 Hardware Programming

The next step is to program the synthesized bitstream and compiled elf on the FPGA(s). . This is done from the GUI by clicking the Program FPGA(s) button on the Synthesis toolbar or selecting it from the Synthesis menu.

When activated, the programming configuration dialog will be displayed. This dialog can also be accessed from Design->Configure Programming menu, or the configure programming button on the toolbar.



From this dialog, the cable type and port for each board/FPGA can be configured. *Note, in the above dialog the Program Board(s) button is disabled. When this dialog is activated from the Synthesis menu, the button is enabled to allow the user to program the board(s).*

Once the cable information has been set, clicking the Program Board(s) button will activate the JTAG Programmer tool to configure the boards.