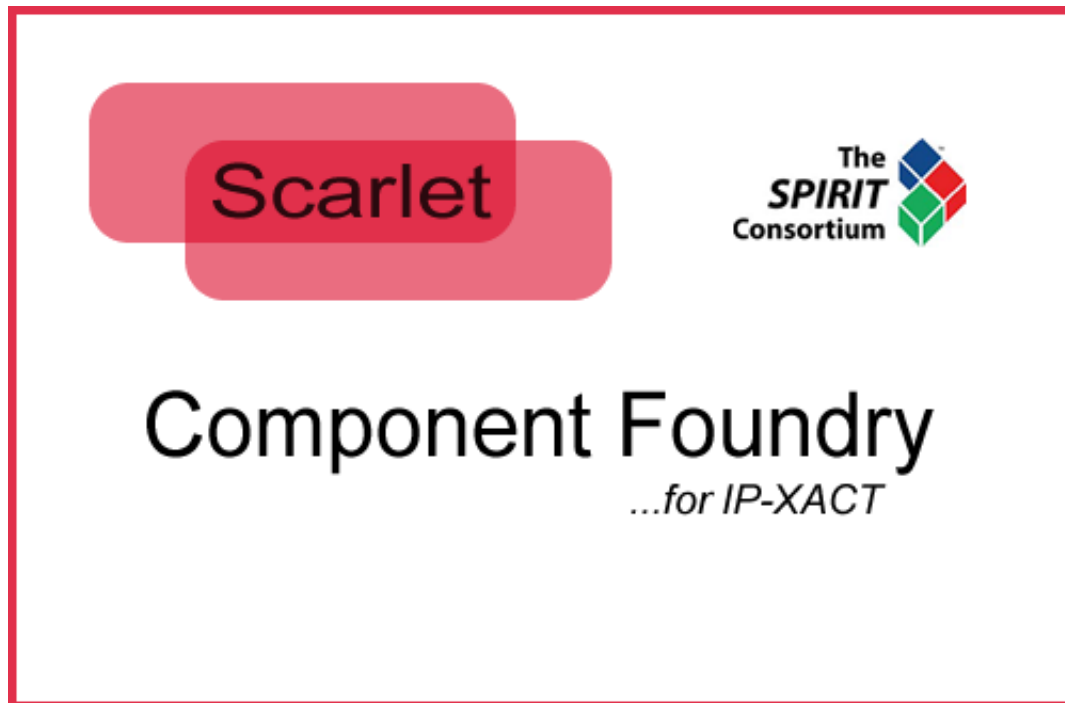

Verilog to IP-XACT Conversion Quick User Guide



Documentation Reference: QUG-102
Revision: 1.0
Publication Date: December 2007

Copyright

© Copyright Scarlet Code Ltd 2007. All rights reserved.

Notice

Neither the whole nor any part of this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder.

This document is intended only to assist the reader in the use of the product(s) it relates to. Scarlet Code Ltd shall not be liable for any loss or damage arising from the use of any information in this document, or for any error or omission in such information, or for any incorrect use of the product(s).

Trademarks

Words and logos marked with ® or ™ are registered trademarks or trademarks owned by Scarlet Code Ltd, except as otherwise stated below in this proprietary notice. Other brands and names mentioned herein may be the trademarks of their respective owners.

Web address

<http://www.scarletcode.co.uk>

Publication History

Revision	Date	Comment
1.0	December 2007	Initial version

Contents

1. ABOUT THIS DOCUMENT	4
1.1 FEATURES.....	4
1.2 PREREQUISITES.....	4
1.3 CREATING AN EXAMPLE PROJECT.....	5
1.4 STARTING THE VERILOG WIZARD	5
1.5 THE VERILOG WIZARD	6
1.6 RE-RUNNING THE VERILOG WIZARD	8
1.7 BUS INTERFACE CREATION	8
1.7.1 RESETTNG THE COMPONENT FOUNDRY PERSPECTIVE	9

1. About this document

This document describes the use of the Verilog to IP-XACT Parser implemented within Scarlet Code Component Foundry.

Verilog to IP-XACT parser is used to create an IP-XACT Component with the top level signals found from processing a hierarchy of Verilog files. Furthermore, by analysing these signals and comparing them against the available Bus Definitions, the parser will also instantiate appropriate Bus Interfaces. In addition, the parser will create a fully populated *FileSet* for the Verilog files within the Component.

The Scarlet Code Component Foundry Verilog to IP-XACT Parser provides the user with a very significant tool for improving the modelling accuracy of IP-XACT files, and greatly reduces their maintenance effort.

1.1 Features

Features of the Scarlet Code Component Foundry Verilog to IP-XACT Parser include:

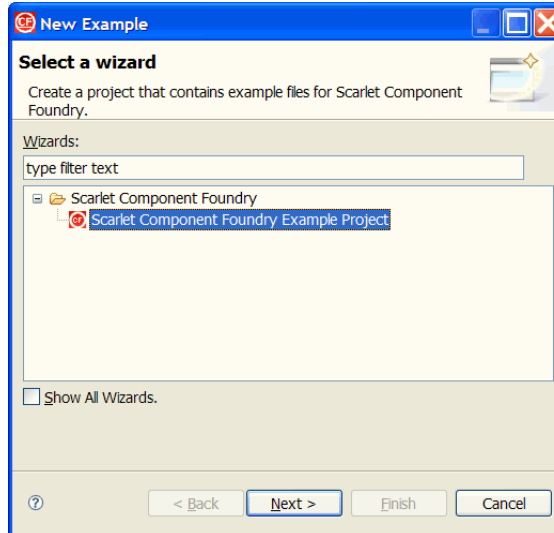
- ❖ Support for both the Verilog-95 & Verilog-2001 standards
- ❖ Handling of the entire module hierarchy and location of the *top* module
- ❖ Builds the top-level signal list
- ❖ Identifies *all* modules within a design
- ❖ Builds a complete file list with *all* include files
- ❖ Handles `'defines`
- ❖ Handles parameterisation
- ❖ Improved IP-XACT accuracy
- ❖ Reduced IP-XACT maintenance effort

1.2 Prerequisites

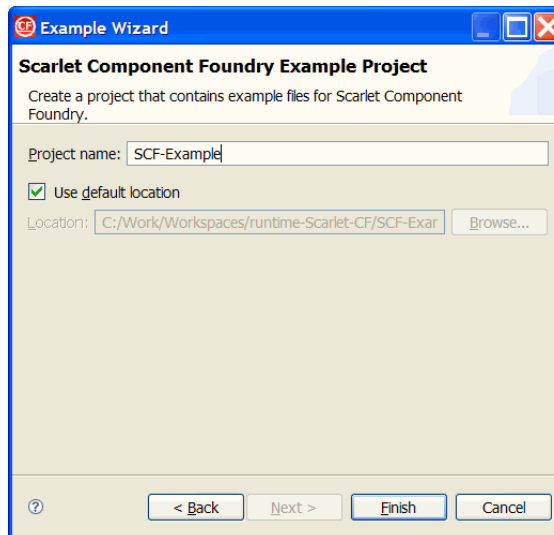
Scarlet Code Component Foundry Verilog to IP-XACT Parser is available in Scarlet Code Component Foundry version 1.2.0 or higher. This document assumes that a licensed copy of a suitable version of Scarlet Code Component Foundry is installed.

1.3 Creating an Example Project

If you wish to try out the wizard, but do not have any Verilog files to hand, you can create a project containing sample Verilog files. To do this select, *File – New – Example...* and then select *Scarlet Component Foundry Example Project* in the *New Example* wizard.



Select *Next*, then enter or accept the *Project name* and select *Finish*.



The new project will be created in your workspace.

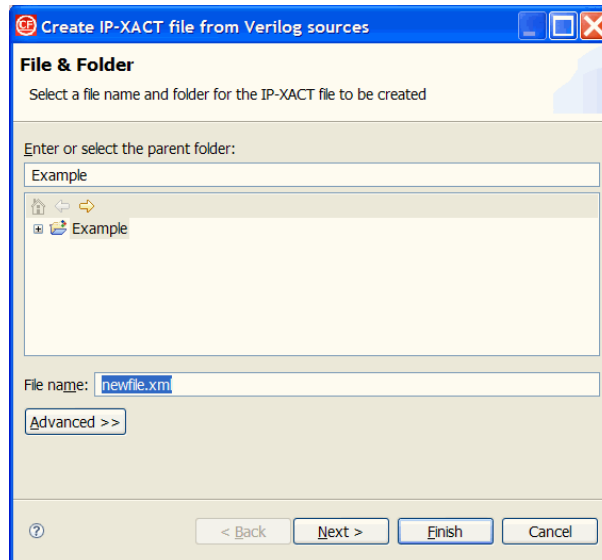
1.4 Starting the Verilog Wizard

The wizard that drives the Verilog to IP-XACT Parser creates a new IP-XACT file from the Verilog source. This wizard is accessed by selecting *File – New – Create IP-XACT from Verilog...* If this option is not present, check you have Scarlet Code Component Foundry version 1.2.0 or higher installed (select *Help – About Scarlet IP-XACT Component Foundry*), alternatively the menu option may not be shown if you have updated from an earlier version. If the latter is the case, you may need to reset the

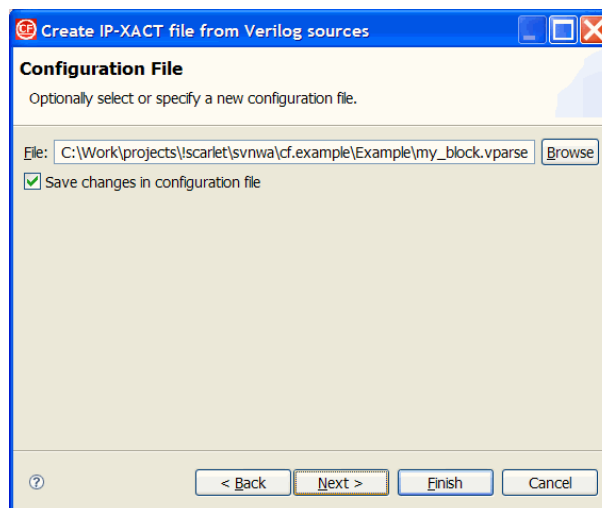
Component Foundry perspective (see *Resetting the Component Foundry Perspective*), or you can find it by selecting *File – New – Other...* and selecting the item from the IP-XACT tree section in the *New* wizard that appears.

1.5 The Verilog Wizard

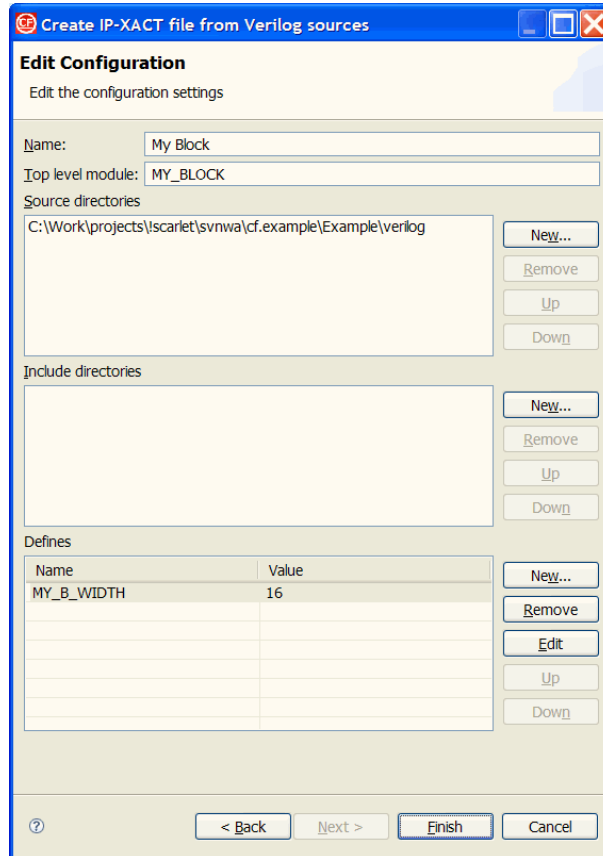
The Verilog wizard comprises a number of pages of options. The first page allows you to select the name and location of the IP-XACT Component file that will be created by the Verilog parser. Select a project folder, enter a file name and press *Next*.



On the second wizard page, you may specify an optional Verilog Parser configuration (`.vparsed`) file. This file holds the Verilog Parser settings and is useful if you want to re-run the wizard again later. If you wish to create this file, enter a file name in the *File* field. You should also ensure the *Save changes in configuration file* is checked. This latter option may be useful if you are re-running the wizard from an existing Verilog Parser configuration (`.vparsed`) file, but do not want the file to be updated.

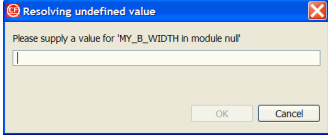
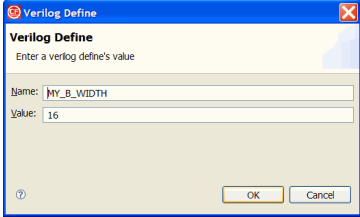


Select *Next*. The final wizard page will be displayed.

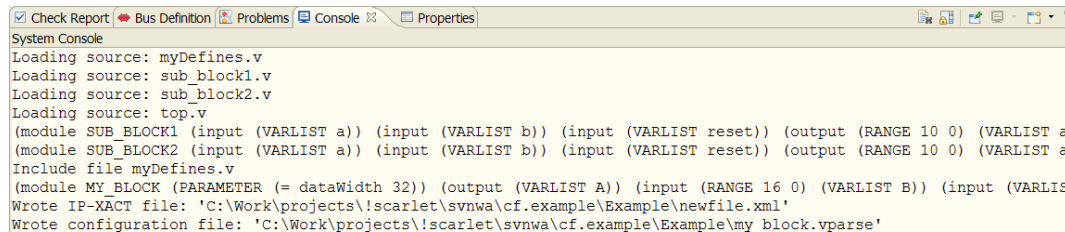


On this page of the wizard you specify the Verilog Parser configuration settings. The only mandatory fields are those for the *Name* and *Source directories*.

Field	Comment
Name	Enter a name to describe or identify the configuration. This name is useful if you have reloaded a configuration.
Top level module	This is the name of the top level module. If it is not known, leave this field blank and the Verilog Parser will prompt for it when it has determined the possible choices: <div data-bbox="750 1423 1110 1640" data-label="Image"> </div>
Source directories	Use the <i>New...</i> button adjacent to this field to browse for a directory containing your Verilog sources. (You only need to specify the root directory as sub-directories are automatically scanned).
Include directories	Use the <i>New...</i> button adjacent to this field to browse for a directory containing any additional Verilog include files.

Field	Comment
	<p>This table contains <i>name-value</i> pairs to resolve missing 'define values in the Verilog source. If a value is required by the Verilog Parser, but is not specified either in the source or here, then the parser will prompt for the value interactively, and the value will be automatically added to the configuration:</p> 
Defines	<p>To add a new value to the <i>Defines</i> table, use the <i>New...</i> button. Similarly to edit an existing value, use the <i>Edit</i> button:</p> 

Finally, to start the Verilog Parser, select *Finish*. A *Console* view will be opened that shows the parser's log.



```

System Console
Loading source: myDefines.v
Loading source: sub_block1.v
Loading source: sub_block2.v
Loading source: top.v
(module SUB_BLOCK1 (input (VARLIST a)) (input (VARLIST b)) (input (VARLIST reset)) (output (RANGE 10 0) (VARLIST ac
(module SUB_BLOCK2 (input (VARLIST a)) (input (VARLIST b)) (input (VARLIST reset)) (output (RANGE 10 0) (VARLIST ac
Include file myDefines.v
(module MY_BLOCK (PARAMETER (= dataWidth 32)) (output (VARLIST A)) (input (RANGE 16 0) (VARLIST B)) (input (VARLIST
Wrote IP-XACT file: 'C:\Work\projects\!scarlet\svnwa\cf.example\Example\newfile.xml'
Wrote configuration file: 'C:\Work\projects\!scarlet\svnwa\cf.example\Example\my_block.vparse'

```

As the parser processes the data, it may prompt for input as described in the previous table.

Following a successful parse of the Verilog data, the newly created IP-XACT Component will be opened in an editor window.

1.6 Re-running the Verilog Wizard

If you have saved the Verilog Parser's configuration settings in a `.vparse` file, then you may easily re-run the wizard using these settings by right-clicking on the `.vparse` file and selecting the *Create IP-XACT from Verilog...* option.

1.7 Bus Interface Creation

The Verilog to IP-XACT parser analyses the top level signals in the Verilog source and creates corresponding model signals in the new IP-XACT Component file. In addition the Verilog to IP-XACT parser attempts to match these signals against the available Bus Definitions and will create Bus Interfaces where a match can be found. Signals may be matched by more than one Bus Definition, and in this case, more than one Bus

Interface may be created. If this is the case, you should delete any unwanted additional Bus Interfaces in the editor.

1.7.1 Resetting the Component Foundry Perspective

Should you wish to reset the *Component Foundry perspective* (in order to reset the menu items), right-click on the Component Foundry perspective button and select *Reset*. Resetting the perspective does, however, restore the editors and views to their default sizes and positions.

