Synopsys®, Inc.
700 East Middlefield Road
Mountain View, CA 94043 USA
Phone: (U.S.) +1 650.584.5000
Website: www.synopsys.com

Synplify Pro® for Microsemi Edition Release Notes

Version H-2013.03M-SP1-1, October 2013

Publication Version 01

Release Note Topics

About the H-2013.03M-SP1-1 Release

This H-2013.03M-SP1-1 release includes software features and enhancements for the Synplify Pro® Microsemi Edition product. For the complete summary of features and enhancements contained in this release, see H-2013.03M-SP1 Feature and Enhancement Summary, on page 2.

H-2013.03M-SP1 Feature and Enhancement Summary

The following table highlights the H-2013.03M-SP1 features:

Feature	Description
DOTP support	DOTP mode is supported in MACC block inference for SmartFusion2 and IGLOO2. See *DOTP Support* on page 3
Incremental Flow synthesis	Incremental flow synthesis is supported for SmartFusion2 and IGLOO2 devices. See *Incremental Flow Synthesis* on page 4
Verilog Library File Extensions	Support for Verilog library file extensions has been added to the synthesis tools. See *Library Extensions for Verilog Library Files* on page 5.
Constraint Checker for Query Commands	Use the check_fdc_query Tcl command to run the constraint checker for constraints using the get_* and/or all_* query commands. See *New check_fdc_query Command* on page 6.
Archive Utility	The archive utility FTP server location has been updated.
Technical Resource Center	The Technical Resource Center options have been removed from the synthesis tools. **Note:** Use SolvNet to access new software releases available and for product information.

H-2013.03M-1 Feature and Enhancement Summary

The following table highlights the H-2013.03M-1 features:

Feature	Description
New Device Support	This release provides logic synthesis support for Microsemi IGLOO2 devices.
Triple Modular Redundancy (TMR) on Registers support for SmartFusion2 and IGLOO2	TMR support on registers triplicates registers and adds voter logic at the output. See the online help or reference.pdf >Synthesis Attribute and Directive Syntax >Summary of Attributes and Directives > syn_radhardlevel Attribute.
Three Port RAM inference support for SmartFusion2 and IGLOO2	RAM64x18 is a 3-port memory providing one Write port and two Read ports. See the online help or reference.pdf >Designing with Microsemi >Microsemi Components >Microsemi RAM Implementations>SmartFusion2.
Expanded Support for Synopsys sdc Commands	The synthesis tool now supports various get* object query commands and collection manipulation commands. See the online help or reference.pdf->Tcl Find, Expand, and Collections->Object Query Commands and Synopsys Standard Collection Commands.
Expanded SystemVerilog Support	You can now: • Specify the wire declaration for multi-dimensional user-defined arrays. • Use the inside operator to indicate set membership with a case statement. See the online help or reference.pdf->SystemVerilog Language Support->Data Declarations->Nets and Operators and Expressions->Set Membership Case Inside Operator.
Log File and Reporting Enhancements	The navigation panel in the log viewer now consolidates and links directly to reports from various stages of the design, like compiler reports, pre-mapping reports, mapping reports, and place-and-route reports. See the online help or reference.pdf->User Interface Commands->View Menu->View Log File Command.

DOTP Support

MACC block, when configured in DOTP mode, has two independent signed 9-bit x 9-bit multipliers followed by addition. The sum of the dual independent 9x9 multiplier (DOTP) result is stored in the upper 35 bits of the 44-bit output. In DOTP mode, the MACC block implements the following equation:

P = D + (CARRYIN + C) + 512 * ((AL * BH)+ (AH * BL)), when SUB = 0

P = D + (CARRYIN + C) - 512 * ((AL * BH) + (AH * BL)), when SUB = 1

Below is an example RTL which infers MACC block in DOTP mode after synthesis:

```
module dotp_add_unsign_syn ( ina, inb, inc, ind, ine, dout);
```

```
parameter widtha = 6;
parameter widthb = 7;
parameter widthc = 7;
parameter widthd = 8;
parameter widthe = 30;
parameter width_out = 44;

input [widtha-1:0] ina;
input [widthb-1:0] inb;
input [widthc-1:0] inc;
input [widthd-1:0] ind;
input [widthe-1:0] ine;
output reg [width_out-1:0] dout;

always @(ina or inb or inc or ind or ine)  begin

       dout     <= (ina * inb) + (inc * ind) + ine ;

end

endmodule
```

MACC block does not support DOTP mode when:

- Width of the multiplier inputs is greater than 9-bits for signed.
- Width of the multiplier inputs is greater than 8-bits for unsigned.
- Width of the non-multiplier inputs is greater than 36-bits.

Incremental Flow Synthesis

The Synplify Pro tool provides timestamps for each manual compile point in the *_partition.tcl file. The timestamps can be used to check if the compile point was re-synthesized in an incremental run of the tool.

The flow is as follows:

1. Define compile point constraint on the modules in the design. For example:

   ```
   define_compile_point          {<viewname>} -type {locked, partition} -cpfile {}
   ```

2. Run through the normal synthesis flow. Synplify Pro writes timestamps for each compile point in the *<designname>*_partition.tcl file. For example:

   ```
   set_partition_info -name <partition_name> -timestamp <timestamp>
   ```

   With incremental Synplify Pro run, only affected compile points display new timestamps, unaffected compile points retain the old timestamps.

   The Compile Point Summary report is available in the .srr file.

Library Extensions for Verilog Library Files

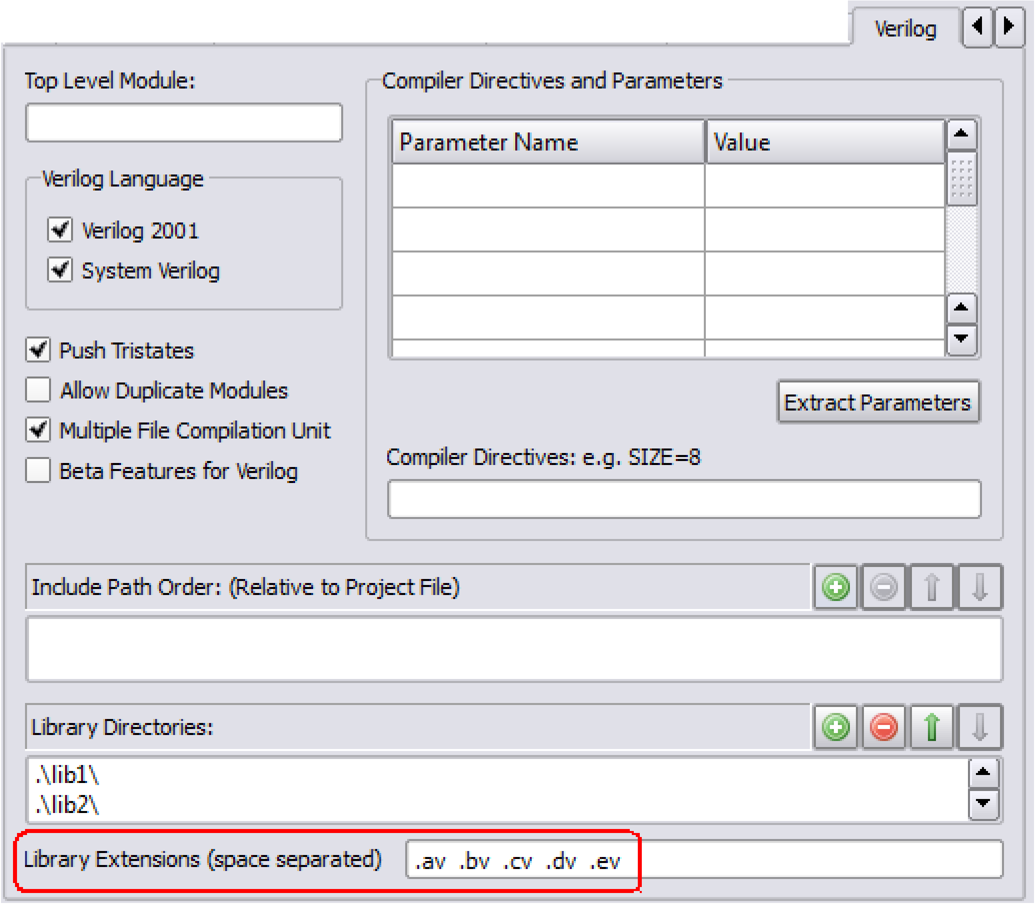
Library extensions can be added to Verilog library files included in your design for the project. When you provide search paths to the directories that contain the Verilog library files, you can specify these new library extensions as well as the Verilog and SystemVerilog (`.v` and `.sv`) file extensions.

To do this:

1. Select the Verilog tab of the Implementation Options panel.
2. Specify the locations of the Library Directories for the Verilog library files to be included in your design for the project.
3. Specify the Library Extensions.

   Any library extensions can be specified, such as `.av`, `.bv`, `.cv`, `.xxx`, `.va`, `.vas` (separate library extensions with a space).

   The following figure shows you where to enter the library extensions on the dialog box.



Top Level Module:

Verilog Language

☑ Verilog 2001

☑ System Verilog

☑ Push Tristates

☐ Allow Duplicate Modules

☑ Multiple File Compilation Unit

☐ Beta Features for Verilog

Compiler Directives: e.g. SIZE=8

Library Directories:

.\lib1\
.\lib2\

Library Extensions (space separated) .av .bv .cv .dv .ev

Verilog
Compiler Directives and Parameters
Parameter Name
Value
Extract Parameters
Include Path Order: (Relative to Project File)

The Tcl equivalent for this example is the following command:

```
set_option -libext .av .bv .cv .dv .ev
```

4. After you compile the design, you can verify in the log file that the library files with these extensions were loaded and read. For example:

```
@N: Running Verilog Compiler in SystemVerilog mode
@I::”C:\dir\top.v"
@N: CG1180 :"C:\dir\top.v":8:0:8:3|Loading file C:\dir\lib1\sub1.av from
specified library directory C:\dir\lib1
@I::"C:\dir\lib1\sub1.av"
@N: CG1180 :"C:\dir\top.v":10:0:10:3|Loading file C:\dir\lib2\sub2.bv from
specified library directory C:\dir\lib2
@I::"C:\dir\lib2\sub2.bv"
@N: CG1180 :"C:\dir\top.v":12:0:12:3|Loading file C:\dir\lib3\sub3.cv from
specified library directory C:\dir\lib3
@I::"C:\dir\lib3\sub3.cv"
@N: CG1180 :"C:\dir\top.v":14:0:14:3|Loading file C:\dir\lib4\sub4.dv from
specified library directory C:\dir\lib4
@I::"C:\dir\lib4\sub4.dv"
@N: CG1180 :"C:\dir\top.v":16:0:16:3|Loading file C:\dir\lib5\sub5.ev from
specified library directory C:\dir\lib5
@I::"C:\dir\lib5\sub5.ev"
Verilog syntax check successful!
```

New check_fdc_query Command

Runs the constraint checker for constraints using the get_* and/or all_* query commands specified in the timing constraint file for the project.

Syntax

check_fdc_query [**-full_check**]

Arguments and Options

-full_check

Runs the full constraint checker before checking the query commands. The default is to run the check_fdc_query command without this option.

When this option is *not* specified, only the constraint syntax checker is run. Its advantage is reducing runtime significantly, since most objects being searched are found in pre-mapping and do not require full mapping to be run. However, this option does not find bit-blasted registers and objects using the advanced -filter @property == commands, where the property is created or applied during mapping because it requires optimizations such as register replication.

For example, if a 4-bit RAM output is targeted using the get_cell command, the constraint check search during:

- Pre-mapping (default) finds ram_out [3:0]
- Full mapping (-full_check option) finds
  - ram_out [3]
  - ram_out [2]
  - ram_out [1]
  - ram_out [0]

Description

The check_fdc_query command reads the `fdc` constraint file of the current project file and runs the constraint checker for any of the following object query commands that are used with FDC constraints:

all_* Commands	get_* Commands
all_clocks	get_cells
all_inputs	get_clocks
all_outputs	get_nets
all_registers	get_pins
	get_ports

The report that is generated allows you to get feedback on how these query commands are applied. This ensures these commands are used properly with constraints in the constraint file.

Note: Collections created with define_scope_collection, find, and expand are not reported by this Tcl command. You can already check these SCOPE collections in the HDL Analyst and the SCOPE interface. Additionally, the report is not generated for the define_io_standard constraint.

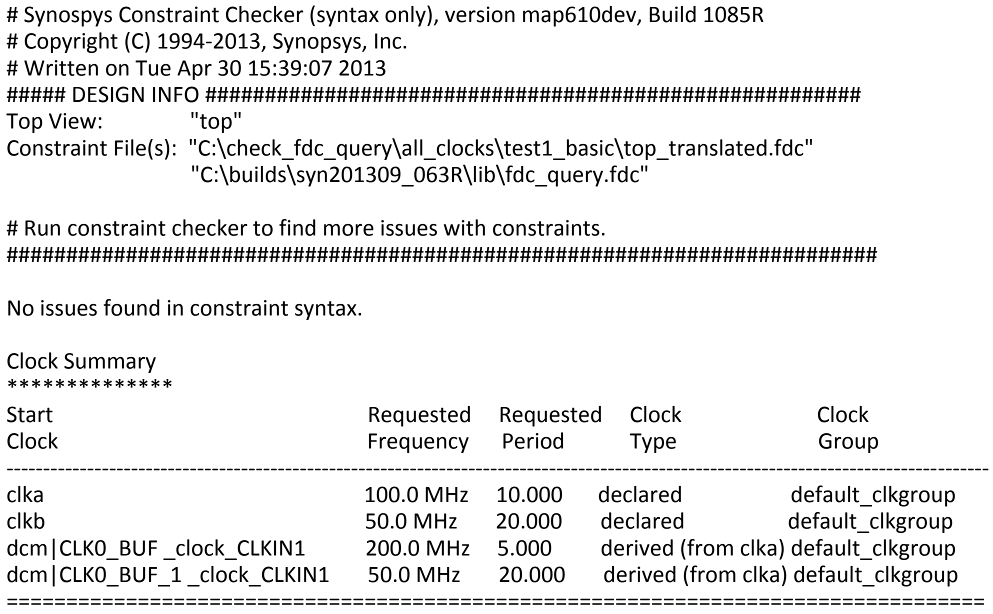
Example

Invoke check_fdc_query from the Tcl command line for the project. When the command finishes, the object query command results are written to the *projectName*_cck_fdc.rpt file that opens in the GUI. You may need to run the constraint checker (Run->Constraint Check) to find more issues with constraints.

The following example shows the results of running the constraint checker in the *projectName*_cck_fdc.rpt file.

```
FDC query commands results
**************************
################################
# 1019 : set_multicycle_path 2 -from [get_cells -hier {*[4]}]
# line 175 in  : C:/check_fdc_query/all_clocks/test1_basic/top_translated.fdc
Results of query command: get_cells -hier {*[4]}
    (none)
################################
# 1027 : set_multicycle_path 3 -to [all_clocks]
# line 196 in  :
C:/check_fdc_query/all_clocks/test1_basic/top_translated.fdc
Results of query command: all_clocks
    clka
    clkb
    dcm|CLK0_BUF_clock_CLKIN1
    dcm|clk0_i_clock_CLKIN1
    dcm|CLK0_BUF_1_clock_CLKIN1
```

The syntax checker reports the object query commands and any issues it found and writes them to the *projectName*_scck.rpt file.



```
# Synospys Constraint Checker (syntax only), version map610dev, Build 1085R
# Copyright (C) 1994-2013, Synopsys, Inc.
# Written on Tue Apr 30 15:39:07 2013
##### DESIGN INFO ##########################################################
Top View:           "top"
Constraint File(s):  "C:\check_fdc_query\all_clocks\test1_basic\top_translated.fdc"
                    "C:\builds\syn201309_063R\lib\fdc_query.fdc"

# Run constraint checker to find more issues with constraints.
###############################################################################

No issues found in constraint syntax.

Clock Summary
**************
Start                                    Requested    Requested    Clock                  Clock
Clock                                    Frequency    Period       Type                   Group
------------------------------------------------------------------------------------------------------------------------------------------
clka                                     100.0 MHz    10.000       declared               default_clkgroup
clkb                                     50.0 MHz     20.000       declared               default_clkgroup
dcm|CLK0_BUF _clock_CLKIN1               200.0 MHz    5.000        derived (from clka) default_clkgroup
dcm|CLK0_BUF_1 _clock_CLKIN1             50.0 MHz     20.000       derived (from clka) default_clkgroup
==========================================================================================
```

Platforms

This section includes platform support for the Synopsys FPGA synthesis products. The software is supported on the platforms and operating systems listed below:

Windows (x86/x64)	• Windows 7 Professional or Enterprise (32/64-bit) • XP Professional SP2 or later (32/64-bit) • Vista Enterprise or Business (32/64-bit). This is the last release to support this platform.
Linux (x86/x64)	All Linux platforms require 32-bit compatible libraries. • Red Hat Enterprise Linux 4/5/6 (32/64-bit). This is the last release to support RHEL 4 and 32-bit RHEL 6. • SUSE Linux Enterprise 10/11 (32/64 bit)

Required Operating System Patches

Running this software requires that the Linux operating system include specific patches. To determine whether your operating system requires patches, refer to the following procedure.

Checking the Installed Patches

All Linux-based FPGA synthesis applications include a script (syn_system_check) that is designed to check patches that have been installed and the patches that need to be installed or updated.

To use this script:

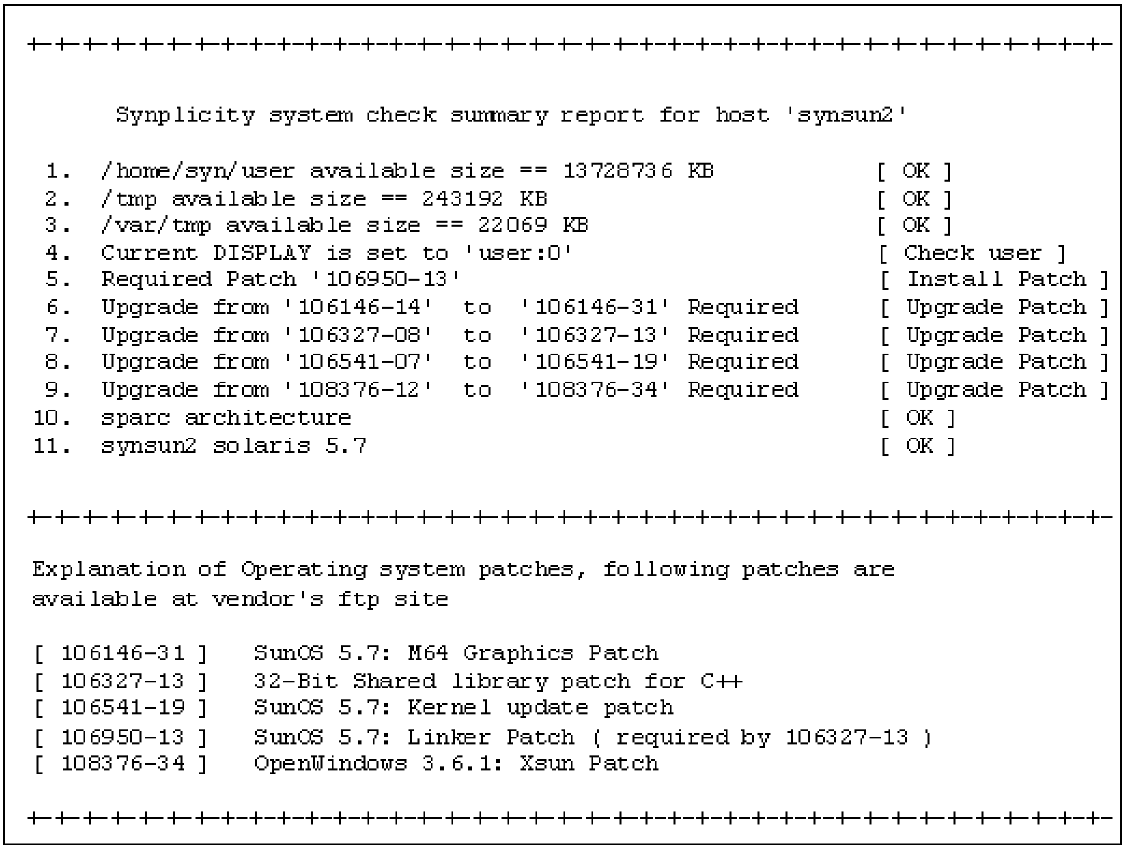
1. Install the product software.
2. Run the script by entering the following command in a shell:

   /*install_dir*/*product_version*/bin/syn_system_check

3. The script runs and generates a system check summary report that lists the patches and patch status (OK, Install, or Upgrade).
4. Consult the display and install or update any of the patches indicated.

Example

The following is a sample report.



```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-


      Synplicity system check summary report for host 'synsun2'

  1.  /home/syn/user available size == 13728736 KB            [ OK ]
  2.  /tmp available size == 243192 KB                        [ OK ]
  3.  /var/tmp available size == 22069 KB                     [ OK ]
  4.  Current DISPLAY is set to 'user:0'                      [ Check user ]
  5.  Required Patch '106950-13'                              [ Install Patch ]
  6.  Upgrade from '106146-14'  to  '106146-31' Required      [ Upgrade Patch ]
  7.  Upgrade from '106327-08'  to  '106327-13' Required      [ Upgrade Patch ]
  8.  Upgrade from '106541-07'  to  '106541-19' Required      [ Upgrade Patch ]
  9.  Upgrade from '108376-12'  to  '108376-34' Required      [ Upgrade Patch ]
 10.  sparc architecture                                      [ OK ]
 11.  synsun2 solaris 5.7                                     [ OK ]


+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-

 Explanation of Operating system patches, following patches are
 available at vendor's ftp site

 [ 106146-31 ]   SunOS 5.7: M64 Graphics Patch
 [ 106327-13 ]   32-Bit Shared library patch for C++
 [ 106541-19 ]   SunOS 5.7: Kernel update patch
 [ 106950-13 ]   SunOS 5.7: Linker Patch ( required by 106327-13 )
 [ 108376-34 ]   OpenWindows 3.6.1: Xsun Patch

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Documentation

The following documents are included with the Synopsys FPGA synthesis products. All documents can be accessed through the online help (HTML), and as PDF documents. See *Accessing Online Help* on page 10, and *Accessing PDF Documents* on page 10 for information on how to access the documents.

Document	Access
User Guide	Online help, context-sensitive help, PDF
Reference Manual	Online help, context-sensitive help, PDF
Error Messages Guide	Online help, context-sensitive help from the log file or Message viewer, PDF
System Designer User Guide	Online help, context-sensitive help, PDF

Accessing PDF Documents

PDF documents display in Adobe Acrobat Reader. You can download the latest Acrobat Reader at no cost from Adobe's website (www.adobe.com). The PDF files provided are optimized for output to a laser printer, not for viewing online.

You can access the PDF documents in multiple ways, while running the tool or separately:

- From within the tool, with the Help-Online Documents command.
- From outside the tool, through the Documentation tab in SolvNet.
- From outside the tool, as described below:

Linux	From outside the software, select Open Acrobat Reader: `acroread` Open *installDirectory*`/documents/`*`docFile`*
Windows	Start->Programs->Synopsys->FPGA Synthesis H-2013.03M-SP1-1->Documents Then, select the desired document.

If the PDF for the online documentation does not open in the synthesis tool, make sure you are using a recent version of the Acrobat Reader that can be downloaded from Adobe's website. Do not use the View->Full Screen option when viewing documents in Acrobat Reader unless you are sure you want this full magnification. On some applications, this selection takes over the monitor and there is no apparent way to access other running tasks or windows. If you do happen to use the Full Screen option, you can use Ctrl-l to undo it.

Accessing Online Help

This section describes how to access online help and context-sensitive help in the Synplify Pro tool.

If your online help graphics do not display correctly, this is usually because your Display setting is 256-Color. Reset it to 16-bit Color, then reopen the online help.

Accessing Help from inside the tool	
All platforms	Select Help->Help for the online help system. For context-sensitive help, click F1 in a dialog box or window. For context-sensitive help for an error message, click the link in the log file or the Message viewer. If your online help search does not locate error messages, open online help and set the Filtered by field to either Unfiltered or Messages Only.
Accessing Help from outside the tool	
Windows	Select Start->Programs->Synopsys->FPGA Synthesis H-2013.03M-SP1->Help
Linux	Run fpga_help.

Known Problems and Solutions

The following problems apply to supported features in the Synplify Pro tool.

Manually Copy Compile Point Sub-directories to Identify Implementation

When compile points are included in an existing FPGA implementation and a new Identify implementation is created, the compile-point related data is not copied to the Identify implementation.

Solution: Manually copy the sub-directories in the FPGA implementation to the new Identify implementation directory. In addition to the sub-directories, the top-level constraint file and all of the compile-point constraint files for the related modules must be enabled on the Constraints panel of the Implementation Options.

Up-to-date Checking Limitation for Sourced Tcl Files in the SDC File

When a design is run, the up-to-date checking feature automatically determines if a design needs to be re-synthesized or not. This feature can provide significant runtime improvements especially for team designs.

However, when you modify constraints in a Tcl file sourced within the constraints file (`sdc`), the software is not aware of these changes and does not force the design to be re-synthesized.

Solution: This will be fixed in a future release.

Handling State Machines in Different Clock Domains

If a state machine defined in the code feeds sequential elements in a different clock domain, using any encoding value other than the "original" can cause metastability. By default, the synthesis tools choose the optimal encoding value based on the number of states in the state machine. This can introduce additional decode logic that may cause metastability when it feeds sequential elements in a different clock domain.

Solution: As a workaround, use syn_encoding = "original" to guide the synthesis tool for these cases.

Verilog Include Paths Missing from VCS Integration

If Verilog include paths have been added to your Project file, these paths are not automatically added to the VCS script.

Solution: You must add the Verilog include paths manually. To do this, use one of the following workarounds:

- From the Run VCS Simulator dialog box, add `+incdir+<`*`include path`*`>` in the Verilog Compile options field.
- Modify the VCS script file, adding the `+incdir+<`*`include path`*`>` to all or any relevant vlogan commands.

Crossprobing Source Code Files Created with Third-Party Editors

When using source code files created with third-party editors, you sometimes cannot crossprobe to the correct line number in the source file.

Solution: Open the file in the FPGA synthesis tool text editor.

Editing Externally Created Project (prj) Files

If Tcl commands or script files were used to build your project, you might not be able to save this Project file from the synthesis UI in downstream tools, because they contain hard-coded file paths.

Solution: Generally, use the same method to save a project as you did to create the project. In this case, save the project file to an external text editor and not in the project UI.

Digital Line Detect Error Appears (Windows XP)

When launching the synthesis software, a Digital Line Detect error occasionally appears on a Dell PC running Windows XP.

Solution: Download the latest update from Dell Corporation. See the following URL for more information.

http://support.dell.com/support/downloads/format.aspx?releaseid=r84541&c=us&l=en&s=gen&cs

Alternatively, you can deselect the Digital Line Detect entry in your system configuration. To do this:

1. Select Run from the Start menu.
2. Type `msconfig` in the open field and click OK.

   The System Configuration Utility dialog box appears.
3. Select the Startup tab and scroll down to the Digital Line Detect entry.
4. Deselect the checkbox to disable it.

Invoking Tool can be Slow on Linux With Network Problems

Invoking the synthesis tools might be slow when you are experiencing network problems on Linux platforms, such as when a mounted drive is not accessible.

Solution: Delete the `$HOME/.config/Trolltech.conf` file to avoid caching. This might help to invoke the tool faster.

Synopsys, Inc.
700 East Middlefield Road, Mountain View, CA 94043 USA
Phone: +1 650 584-5000 or +1 800 541-7737