Synopsys®, Inc.
690 East Middlefield Road
Mountain View, CA 94043 USA
Website: www.synopsys.com

Synopsys® FPGA Design Microsemi Edition Release Notes

Includes Synplify Pro® and Identify®
Version L-2016.09M-SP1-3, August 2017

Release Note Topics

About the Release

This L-2016.09M-SP1-3 release includes software features and enhancements for the Synplify Pro® and Identify Microsemi Edition products. For the complete summary of features and enhancements contained in this release, see Feature and Enhancement Highlights below.

Feature and Enhancement Highlights

The following table summarizes the supported features and enhancements:

Feature	Description
Identify Features in the L-2016.09M-SP1-3 Release	
New command com cableoptions	The new command enables you to specify a particular FlashPro programmer. This command is used when a computer host has multiple programmers connected to it. See Microsemi FlashPro Programmer Selection, on page 5.
Soft JTAG TAP Controller	The SoftJTAG Controller feature is now fixed and available for use. The new com cableoptions command can be used with the SoftJTAG Controller as well, with a limitation. See Signal Integrity Issues when using FlashPro5 Programmer with SoftJTAG, on page 9 for more information.
Synplify Pro Feature in the L-2016.09M-SP1-1 Release	
Coefficient ROM Packing in DSP Block	The tool is enhanced to support the coefficient ROM packing in DSP block feature with the inference of MACC_PA_BC_ROM macro for the PolarFire device. *Reference->Designing with Microsemi->Microsemi Components->DSP Block Inference*
Synplify Pro Features in the L-2016.09M-SP1 (Linux) & L-2016.09M-SP1-1 (Windows) Release	
Initial Values for Registers (SLE)	For Microsemi technologies, initial values on registers (SLE) are not supported. If initial value is specified for a register in the RTL code, the tool ignores the value and issues a warning. *Reference->Designing with Microsemi->Microsemi Components*
Device Support	The Synplify Pro tool now supports the Microsemi PolarFire device.
RAM Inference for ROM support	The tool supports inference of RAM (RAM64x12 & RAM1Kx20) primitive for ROM. This support is available for PolarFire technology only. *Reference->Designing with Microsemi->Microsemi Components->Microsemi RAM Implementations*
syn_romstyle Attribute Support	Determines how ROM architectures are implemented. *Attribute Reference->Attributes and Directives->Attributes and Directives Summary*

Write Byte-Enable Support for RAM	The Synplify Pro tool is enhanced to support Write Byte-Enable feature for RAMs inferred in non-low power (speed) mode. It is supported for PolarFire and RTG4 technologies only. *Reference->Designing with Microsemi->Microsemi Components->Microsemi RAM Implementations*
Automatic Compile Points (ACP) Support	The tool is enhanced to support Automatic Compile Points (ACP) flow. This option is ON by default for PolarFire technology only. *User Guide->Working with Compile Points->Compile Point Basics*
syn_no_compile_point Attribute Support	Use with the Automatic Compile Point (ACP) feature. The software automatically identifies modules as compile points in the design based on its size, number of I/Os, and hierarchical levels. If you do not want the software to create a compile point for a particular view or module, then apply this attribute. *Attribute Reference->Attributes and Directives->Attributes and Directives Summary*
Compiler Enhancement	Cross-module referencing is supported for generate blocks that use a loop variable in the conditional statement. *Language Support Reference->Verilog Language Support ->Verilog 2001 Support->Cross-Module Referencing->Cross-Module Referencing of Generate Blocks*
Identify Feature in the L-2016.09M-SP1 (Linux) & L-2016.09M-SP1-1 (Windows) Release	
Identify Debugger Stand-alone Installation Package	The Identify debugger executable is packaged and installed separately.
Synplify Pro Features in the L-2016.09M Release	
SLE Enhancement	The synthesis tool is enhanced to support packing the enable signal with higher priority than the reset signal (synchronous), into SLE. *Reference->Designing with Microsemi->Microsemi Components* *->Control Signals Extraction for Registers (SLE)*
RAM64x18, RAM64x18_RT, RAM1K18_RT Enhancements	The Synplify Pro tool is enhanced to support packing the enable signal on the read address register into RAM1K18_RT (A_REN), RAM64x18 (A_ADDR_EN & B_ADDR_EN), and RAM64x18_RT (A_ADDR_EN & B_ADDR_EN). *Reference->Designing with Microsemi->Microsemi Components->Microsemi RAM Implementations*
Wide MUX inference Support	Wide MUXs are implemented using ARI1 primitives and is supported on the SmartFusion2 and RTG4 technologies. *Reference->Designing with Microsemi->Microsemi Components*

Compiler Enhancements	Compiler enhancements include the following: SystemVerilog support for the `begin_keywords and `end_keywords directives. Specifies a pair of directives—`begin_keywords and `end_keywords—to identify keywords reserved within a block of source code, based on a specific version of IEEE Std 1364 or IEEE Std 1800. *Language Support Reference->Verilog Language Support ->Support for Verilog Language Constructs->Compiler Directives* Use the new SYN_COMPATIBLE=DC macro to ensure compatibility of Synopsys tools such as Design Compiler (DC) with the synthesis software. *Command Reference->User Interface Commands ->Implementation Options Command->Compiler Directives and Design Parameters*
New HDL Analyst® Tool	*Beta* A new version of the next-generation schematic analysis tool is enabled by default. To go back to the original HDL Analyst tool, click the button in the upper right of the tool window or deselect the option HDL Analyst->Use New HDL Analyst (Beta). This version includes usability improvements, better performance and support for designs that generate large netlists. *User Guide->Analyzing with HDL Analyst->Working in the Schematic (Beta)* *Beta* The HDL Analyst also uses new Tcl and find commands. *Command Reference->Tcl Commands->analyst* and *Tcl Commands->design*
Launch an Independent Help	Launch the help system independent of the tool, by running *installDirectory*/bin/fpga_help.exe. It is recommended that you use help instead of PDFs, because help is designed as an integrated system and includes additional navigational aids. You can double-click the executable to start it on Windows.
Identify Feature in the L-2016.09M Release	
Identify Graphical User Interface Changes	Minor changes to the graphical user interface include: • The RTL Instrumentor status panel is rearranged and renamed to Control Panel. • The Instrumentor Search dialog box is replaced with the Search panel in the main view, and the Search icon has been removed. The *Identify Instrumentor User Guide* has been updated to reflect these changes.

Feature and Enhancement Description

This section contains descriptions of the features and enhancements.

Microsemi FlashPro Programmer Selection

The com cableoptions option allows you to select one among the multiple FlashPro programmers connected to a common host:

```
com cableoptions Microsemi_BuiltinJTAG_port <string>
```

The string represents the FlashPro programmer's port name.

You can identify the port name and proceed to use the cable option as described below:

1. Start FlashPro.
2. Scan the programmers that are connected to the host and note down the port name (for example—usb32344).
3. Close FlashPro.
4. Start Identify debugger.
5. Define the cable type as:

   ```
   com cabletype Microsemi_BuiltinJTAG
   ```

6. Define the cable option using the FlashPro programmer port name that you identified in Step 2. For example:

   ```
   com cableoptions Microsemi_BuiltinJTAG_port usb32344
   ```

   Note: For Flashpro4 programmer ports, the port name must include the usb prefix, as shown in the example above. Flashpro5 ports on the other hand, must NOT include the prefix. For example:

   ```
   com cableoptions Microsemi_BuiltinJTAG_port S201R1NLS
   ```

7. Check communication with the port using the com check command. If the check is successful, you can start the debugger and debug the design.

Note that you cannot change to a different port by just re-running step 6 with the new port's name. To select a different port, perform the following steps:

1. Stop the server using the jtag_server stop -forced 0 command. If this does not work, use -forced 1.
2. Define the new cable option. For example:

   ```
   com cableoptions Microsemi_BuiltinJTAG_port usb32388
   ```

3. Run com check to check communication with the new port.

Identify Tool Device Support

The Identify tool supports the device families shown in the table below. You must select devices from the synthesis tool, which get passed to the Identify Instrumentor in the synthesis project file. If you specify a library from the synthesis tool that is not supported in the Identify tool, then this results in a “device not supported” message when launching the Identify Instrumentor.

Microsemi
Fusion
IGLOO
IGLOO PLUS
IGLOO2
ProASIC
ProASIC3
ProASIC3E
ProASIC3L
RTG4
SmartFusion
SmartFusion2
Fusion
PolarFire

Recommended Versions of Compatible Tools

The FPGA design tools are tested with specific versions of other compatible Synopsys and third-party tools. The recommended versions of these tools are listed below.

Compatible Versions of Synopsys Tools

The table lists the recommended version for VCS:

Tool	Recommended Version
VCS	L-2016.06-SP1

Platforms

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

Windows	• Windows 10 Professional or Enterprise (64-bit) • Windows 8.1 Professional or Enterprise (64-bit) • Windows 7 Professional or Enterprise (64-bit) • Windows Server 2008 R2 (64-bit) • Windows Server 2012 R2 (64-bit)
Linux	All Linux platforms require 32-bit compatible libraries. • Red Hat Enterprise Linux 5[1]/6/7 (64-bit) • SUSE Linux Enterprise 11/12 (64-bit)

1. Support for Linux Red Hat Enterprise 5 64-bit will be discontinued after release L-2016.09-SP1.

Documentation

The following documents are included with the Synopsys FPGA synthesis product.

Document	Access
User Guide	Online help, PDF
Reference Manual	Online help, PDF
Attribute Reference Manual	Online help, PDF
Command Reference Manual	Online help, PDF
Language Support Reference Manual	Online help, PDF
Messages Reference Manual	Online help
Identify Instrumentor User Guide	Online help, PDF
Identify Debugger User Guide	Online help, PDF
Identify Debugging Environment Reference Manual	Online help, PDF

Known Problems and Solutions

The current known problems in the tool are divided into the following categories:

- FPGA Synthesis Known Problems and Solutions, on page 8
- Identify Tool Known Problems and Solutions, on page 9

FPGA Synthesis Known Problems and Solutions

The following problem applies to supported features in the Synplify Pro product.

False Warning Possible with Message MT481

Warning message MT481 (Unable to generate clock *clockName* because no clock was found on its master pin. Make sure that a valid pin or port is defined by the -source option for the create_generated_clock command) can be incorrectly reported when the name of a derived clock is defined by its associated create_generated_clock constraint using the -name {*clkName*} *object* format.

Solution: Warnings that result from derived clocks can be ignored. These clocks, which falsely trigger the warning, are not used and are subsequently removed by the mapper. The reporting problem is corrected in the next release.

Identify Tool Known Problems and Solutions

The following problems are specific to the Identify instrumentor and Identify debugger tools.

Signal Integrity Issues when using FlashPro5 Programmer with SoftJTAG

Signal integrity issues are observed when working with SoftJTAG using FlashPro5 programmer. This results in the Identify debugger being unable to communicate with the device.

Solution: Use FlashPro4 programmer instead.

Incremental Instrumentation from Previous Release Cannot be Used

Attempting to open an incremental instrumentation created from a previous Identify release results in an assertion error.

Solution: The incremental instrumentation from the previous release cannot be used, and a new instrumentation must be redefined using this new release.

Unable to Launch the GDKWave Viewer from the Debugger

Occasionally when launching the GTKWave viewer from Linux, the viewer fails to open with the following message:

ERROR: couldn't execute "*installPath*/bin/gtkwave/gtkwave": no such file or directory

Solution: Restart both the Synplify tool and the debugger. This problem is scheduled to be addressed in a future release.

Instrumentor Can Become Unresponsive While Using Multiple Implementations

When using multiple implementations, selecting a different instrumentation from the Instrumentations Select field at the bottom left of the RTL instrumentor view can cause the instrumentor to become unresponsive.

Solution: Select the desired implementation from the project view in the synthesis tool, then invoke the instrumentor rather than attempting to select a different instrumentation directly from the Instrumentations Select field.

Issue Running Identify Instrumentor and Debugger on Different Platforms

When using real-time debugging with the Identify instrumentor running on a Linux platform and the Identify debugger running on a Windows platform, an error is reported when scanning the logic analyzer stating that the remote copy (RCP) could not be executed.

Solution: Run the Identify debugger from the Linux platform.

Pod Assignments not Displayed After Execution of the Logic Analyzer Command

When using real-time debugging, the iice assignments report command fails to display any pod assignments after successful execution of the assignpod and submit options of the logicanalyzer command.

Solution: This problem is scheduled to be addressed in a future release.

Trigger Position May be Incorrect for Data Compression with Cross-Triggering

When using data compression with cross-triggering enabled, the trigger position is incorrect for both the internal_memory (BRAM) and hapssram (SRAM) buffer type settings (the data is still valid).

Solution: This problem is scheduled to be addressed in a future release.

Context-Sensitive Help May not Display Correct Help Page on Linux

When using context-sensitive help (F1) for the Identify tool on Linux, help does not open to the expected page.

Solution: Use the table of contents, global index, or the online help search mechanism to access the correct help page.

Limitations

The current limitations in the tool are divided into the following categories:

- FPGA Synthesis Limitations, on page 10
- Identify Tool Limitations, on page 11

FPGA Synthesis Limitations

The following limitations apply to supported features in the Synplify Pro product.

Page Could Not Be Found Message when Invoking Online Help

When online help is first invoked, it creates a cached version of the compiled help file in a local hierarchy to allow you to save preferences, bookmarks, and full-text search information. This cached version records the path to the installed version. If the same product version is subsequently re-installed in a new directory, invoking online help displays a message, "The page could not be found," because the cached version does not recognize the path to the re-installed product.

Solution: Go to the platform-specific directory with the cached help files:

Windows:
C:\Users*username*\AppData\Local\assistant\Synopsys\Synplify\

Linux:
~/.local/share/data/assistant/Synopsys/Synplify/

Delete any “online*” directories from the cache directory.
Restart help. This creates a new cache, and correctly displays the online help.

GUI Processing Can Fail on Windows 7 for the Synthesis Tool

The synthesis tool GUI might intermittently stop responding on Windows 7.

Solution: To resolve this issue, apply the hotfix from Microsoft by going to suppport.microsoft.com/kb/2718841/.

Online Search Does Not Handle Hyphens as Expected

If the search term includes a hyphen (for example, byte-enable), search in online help does not produce the search hits you expect, because it searches for byte and enable. This limitation does not affect underscores.

It is also limited to online help search and does not affect search in PDF documents.

Solution: Here are some workarounds:

- Basic Search: Use the \ character before the hyphen to escape the hyphen
- Try the index

- Basic Search: Try using the * wildcard
- Basic Search, and Advanced Search with exact term: Try the term with a space in place of the hyphen

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 GUI 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 GUI.

Identify Tool Limitations

The following limitations are specific to the Identify instrumentor and Identify debugger tools.

Verilog/SystemVerilog Limitations When Importing Signals from Verdi

The following Verilog/SystemVerilog language limitations are present when importing signals directly from the Verdi® platform:

- Enums with syn_enum_encoding attribute are not supported for instrumentation and, if present, can impact data expansion.
- Trigger expression settings for unions are either in the form of a serialized bit vector or hex/integer with the trigger bit width representing the maximum available bit width among all union members. Trigger expressions using enum are not possible.
- Generate statements are not supported.
- A limitation exists in the instrumentation of essential signals generated by the Verdi platform because of the naming convention used to represent certain essential signals by the Verdi tool. Instrumentation of such signals cannot be performed automatically using the Verdi getsignals command. The Identify instrumentor issues a warning message when these type of signals are encountered.
- Instrumentation of Interfaces is not supported.

VHDL Limitations When Importing Signals from Verdi

The following VHDL language limitations are present when importing signals directly from the Verdi platform:

- Boolean vector representation in the Identify-generated FSDB is different from the VCS generated FSDB, but does not have any known impact during the data expansion.
- Record elements are represented in reverse order in the Identify-generated FSDB. This reversal does not have any known impact during data expansion.
- Generate statements are not supported.

External Triggering with Imported Triggers Can Cause Excessive Use of the Internal Block RAM

Use of external triggers via the import trigger mechanism causes an excessive use of internal block RAM due to sampling of the trigger as well as the creation of a look-up table. The problem is most notable when the maximum of eight imported triggers is selected.

Solution: Add an extra input to the top-level RTL code and instrument the input as a trigger only.

Synopsys, Inc.
690 East Middlefield Road
Mountain View, CA 94043 USA