Identify® Actel Edition Tool Set Release Notes

Version E-2010.09A, September 2010

Publication Version 01

Contents

About This Release

These release notes describe features and enhancements for the E-2010.09A release of the Identify Actel Edition tool set.

Release Features and Enhancements

The following features and enhancements have been included in this release:

- Updated compiler support
- Synopsys Common Licensing (SCL) support
- Support for the Actel SmartFusion and IGLOO PLUS component libraries
- Partial instrumentation of fields within a record or a structure

Compiler Support

This release of the Identify software supports the instrumentation of designs using the latest constructs from the Synopsys Verilog, SystemVerilog, and VHDL compilers. For information on the applicable constructs, see Chapter 5, *Verilog Language Support*, Chapter 6, *SystemVerilog Language Support*, and Chapter 7, *VHDL Language Support* in the *Reference Manual*; see *Current SystemVerilog Restrictions* on page 3 for a list of the current instrumentation restrictions when using SystemVerilog to instrument a design.

Synopsys Common Licensing

Product licensing for the Identify tool set is controlled by the Synopsys Common Licensing (SCL) software and the Synopsys license daemon (snpslmd). For licensing information, see *Licensing* on page 5.

Actel SmartFusion and IGLOO PLUS Device Support

The Actel SmartFusion and IGLOO PLUS families of FPGA devices are now supported. SmartFusion and iglooplus selections appear in the Device family drop-down menu in the project window, and corresponding command line entries are recognized by the device family Tcl command.

Partial Instrumentation

Partial instrumentation allows individual fields within a record or a structure to be instrumented. For information on using partitial instrumentation in the instrumentor, see *Partial Instrumentation* in Chapter 6 of the user guide; for information on using partial instrumentation in the debugger, see *Displaying Data for Partial Instrumentation* in Chapter 7 of the user guide.

Instrumentation Language Limitations

The following RTL language limitations exist when instrumentating a design:

- Variables in VHDL cannot be instrumented.
- Signals appearing within Block statements in VHDL cannot be instrumented.
- Fields of records in VHDL cannot be instrumented.
- Registers declared within an unlabelled Verilog if-generate statement are not supported (use labelled blocks or declare registers globally).

Current SystemVerilog Restrictions

The following language restrictions exist when using SystemVerilog to instrument a design with the E-2010.09A release:

- .* connection port-instantiation construct is currently not supported
- interface/modport structures for port lists and port connection lists are not currently supported
- package/endpackage is supported although instrumentation within the defined package is not supported
- $unit – compilation units are currently not supported
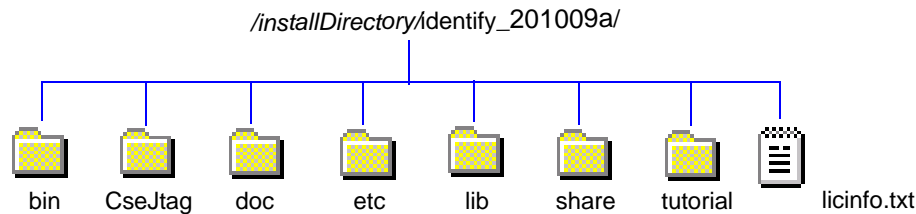- union – the union construct is currently not supported

Please see the *SystemVerilog Language Support* chapter in the *Identify Reference Manual* for complete language support and limitations.

Installation

The directory where Identify is installed is referred to as the *installDirectory*. The installation subdirectory name consists of the Identify product name and an associated version number, in the current case, identify_201009a (for version E-2010.09A). This naming convention permits multiple versions to be installed in the same product directory. The start menus, desktop icons, and uninstall names have an associated version number, in the current case, Identify E-2010.09A. For a list of the compatible Windows-based platforms, see *Platform Support* on page 5.

Downloading the Software

Production versions of the Identify software are installed directly from the downloaded .exe file. Installing the Identify software creates the identify_201009a subdirectory in the installation directory. This directory contains the following files and subdirectories:

*/installDirectory/*identify_201009a/

| bin | CseJtag | doc | etc | lib | share | tutorial | licinfo.txt |

The contents of the files or sub-directories are as follows:

Directory	Contents
bin	Executables for the Identify instrumentor and Identify debugger
CseJtag	Drivers
doc	User documentation
etc	Location for user-defined startup scripts; template included
lib	Program data files
share	Includes the following subdirectories: • contrib – source directory for scripts executed on startup • vhdl – standard vhdl libraries • synthesis – program specific data files • demo_design – the bus_demo design
tutorial	Verilog/VHDL source files and predefined project files for tutorial design.
Text Files	ASCII version of the Synopsys Software License and Maintenance Agreement for Synplicity Software Products.

Licensing

Product licensing for Synopsys products is controlled by the Synopsys Common Licensing (SCL) software and the Synopsys license daemon (snpslmd). The E-2010.09A release of the Identify tool set requires that you use the 11.1 or later release of the SCL software and the included Synopsys license daemon (snpslmd). Note that the Synplicity license daemon (synplctyd) is no longer supported.

Note: Note that re-installation of existing Synopsys FPGA synthesis products is not required when moving to SCL, but you must use the Synopsys license daemon.

The Identify software requires a floating license. To use this software with the Synopsys license daemon, you must install a new license key.

Note: You must refresh your license keys to continue using Synopsys FPGA product releases. Refreshed license keys allow access to existing Synopsys and Synplicity applications and allow you to run the Identify tool set for the E-2010.09A and subsequent releases.

License Variable Settings

After upgrading to the Synopsys Common Licensing software, one of the following variables must be set to point to the license key file:

>   LM_LICENSE_FILE=*port@licenseServerName*
>
>   SNPSLMD_LICENSE_FILE=*port@licenseServerName*

Note that the SYNPLCTY_LICENSE_FILE variable is no longer supported.

Platform Support

The Identify E-2010.09A release is compatible with the following platforms and operating systems:

- Window 7 (32/64-bit); Identify instrumentor only
- Windows XP Professional (32-bit)

Synopsys FPGA Synthesis and Third-Party Tool Compatibility

The E-2010.09A release of the Identify Actel Edition software is compatible with the following Synopsys FPGA and third-party tools on the above platforms and operating systems.

Tool	Recommended Version
Synplify Pro (synthesis)	FPGA D-2009.12A
Actel (place and route)	Libero 9.0
Aldec (simulation)	Active-HDL 7.3-SP1

Always check with your third-party tool vendor to verify current platform support.

Machine Requirements

Machine memory requirements vary according to the size and complexity of your designs. At a minimum, 256 Mbytes of RAM are required (1 Gbyte is recommended). Your machine's virtual memory (swap space) should be set to at least two times the capacity of the RAM.

Location of the .cfg File on Windows

The Windows platforms do not permit applications to write to the C:/Windows directory. Because the Identify tools must update the userprefs.cfg initialization file, this file is written to the following directory location:

```
C:\Documents and Settings\userName\Application Data\Synplicity\Identify
```

Windows Memory Configuration

For all memory configurations, Windows uses a default virtual address space of 4 GBytes; 2 GBytes allocated to user processes (applications) and 2GBytes allocated to the operating system and kernel-mode drivers. On Windows Server 2003 systems and Windows XP Pro systems (with Service Pack 2) that have 1 GByte or more of physical memory, the memory allocation between applications and operating system can be modified to increase the user-process memory allocation to 3 GBytes (and reduce the operating system memory allocation to 1 GByte). For the most up-to-date information for reconfiguring memory allocation on a compatible system, see *How to Set the /3GB Startup Switch in Windows* and related topics on the Microsoft TechNet (http://www.microsoft.com/technet).

Configuring the Windows Firewall for Client Access

Windows XP Service Pack 2 (SP2) and later enhance the security capabilities of the Windows operating system. If you are using a Windows XP machine with an SP2 or later update as your floating license server, you must configure the XP firewall application to allow client access to the floating license ports.

The license manager daemon (lmgrd) and the snpslmd license daemon are separate processes and each requires a separate port to run its associated driver. For floating licenses, you must define a PORT variable for the snpslmd license daemon and then add the ports to the the list of exceptions in the firewall application.
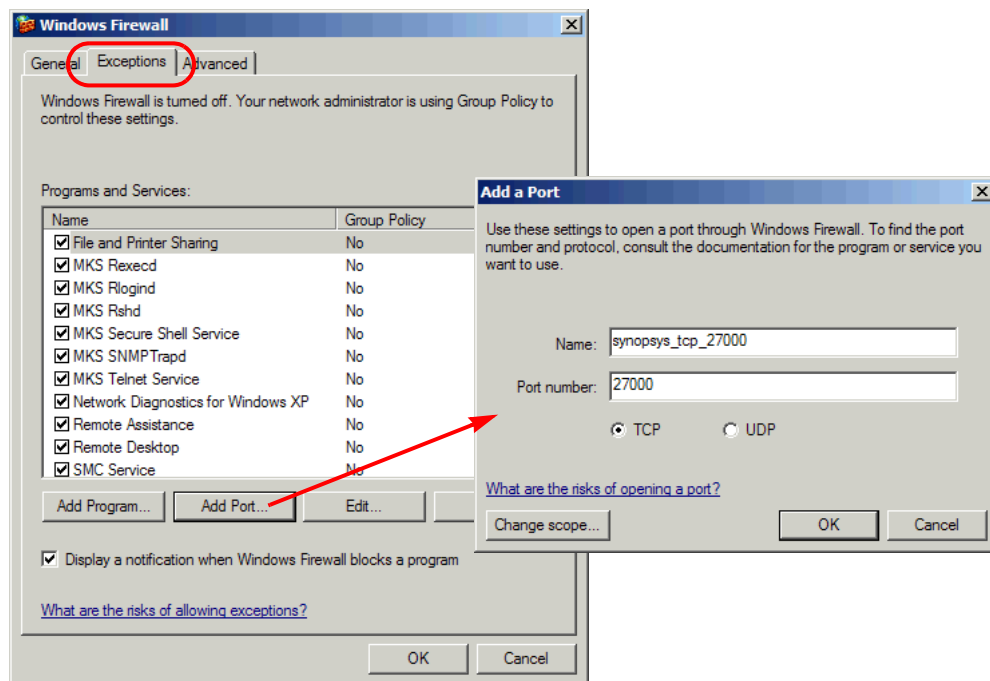
Do the following:

1. Open the license file and edit it to define the variable for the snpslmd license daemon by adding a PORT variable to the VENDOR line.

   The following is an example:

   ```
   SERVER <server_name> FLEXID=9-089Dxxxx 27000
   VENDOR snpslmd c:\synopsys\scl_11.1\windows\bin\snpslmd.exe PORT=27001
   INCREMENT synplifypro snpslmd 2010.12 31-dec-2010 1 \
   ...
   ```

   Next, you must add the port values to the Exceptions list in the Windows Firewall application.

2. From the Control Panel, open the Windows Firewall application.

3. Click the Exceptions tab, and then click the Add Port button to display the Add a Port dialog box.

4. Add the TCP port for the license manager daemon by doing the following in the Add a Port dialog box:

   – In the Name field, specify a name for the TCP port. For example, synopsys_tcp_27000.
   – In the Port number field, enter the assigned TCP port value from the license file. Using the sample license file as an example, enter 27000.
   – Click OK in the Add a Port dialog box to accept the entry and close the dialog box.

5. Specify the port for the snpslmd license daemon as follows:
   – Click the Add Port button to redisplay the Add a Port dialog box, and then set the following options.
   – In the Name field, specify a name for the snpslmd license daemon port. For example, synopsys_port_27001.
   – In the Port number field, enter the assigned PORT value from the license file. Using the sample license file as an example, enter 27001.
   – Click OK in the Add a Port dialog box to accept the entry and close the dialog box.

6. Verify that the port names you assigned are included in the Programs and Services list of the Exceptions pane.

7. Click OK in the Windows Firewall dialog box to exit the Windows Firewall application.

Identify Actel Edition Documentation

The following documentation is included with the Identify Actel Edition E-2010.09A release.

- Identify Actel Edition User Guide
- Identify Actel Edition Reference Manual
- Identify Actel Edition E-2010.09A Release Notes (this document)
- Identify Actel Edition Quick Start Guide
- Identify Actel Edition Quick Tutorial
- Identify Actel Edition Tutorial

Accessing Documents using the Acrobat Reader

The PDF documents contain all the information in the printed document, but are available online. To access the online documentation:

1. If you do not have an Acrobat Reader, download the current version from Adobe's website (www.adobe.com).

2. Open the Acrobat Reader and then open the desired Identify document from the /doc directory under the software installation directory:

Although viewable online, the PDF files provided are optimized for output to a laser printer, not for online viewing, and therefore the quality of graphics and text will be better when printed than when displayed on your screen.

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.

Known Problems and Solutions

The following problems have been identified with the Identify E-2010.09A release:

An incorrect (out of date) adb file can cause the Identify debugger to crash when loading the corresponding activation.

Solution: Either delete the last_run.adb file before opening the debugger or disable Auto-save trigger setting and sample results on the Debugger Preferences dialog box.

Arrays that are indexed by an enumerated type are not supported.

Synthesis fails when instrumenting only one bit in a module instantiated in a generate statement.

Solution: Instrument two or more signals in the instantiated module.

A default license type cannot be set from the shell mode.

Solution: Open the tool in the graphical interface, select the license type making sure that you check the Save as default license type box, and exit the GUI. The tool can now be opened in the shell mode using the default license type.

When using state-machine triggering, partially instrumented buses may not instrument correctly.

Solution: Avoid partially instrumented buses when using state-machine triggering (use full buses or use the complex counter for triggering).

The instrumentor crashes when trying to list more than 1100 signals with Show possible signals.

Solution: Filter the signals before clicking Find.

Unnecessarily attempting to recompile a compiled design in the Identify instrumentor with the Compile button can cause the instrumentor to crash (designs are automatically compiled when the Identify instrumentor is launched from the Synplify Pro synthesis tool, when a project is input from a .prj file, or when a project is opened from a .bsp file).

After launching Identify from Synplify Pro, the following error message is displayed in the Synplify Pro TCL Script window:

```
Job: "Identify Instrumentor UI" terminated with error status: nnn.
```

Solution: Disregard the message.

The -remote_trigger option for the run command is not functional.

The VHDL compiler generates an error (CD455) when a for loop in a generate statement does not start from 0.

Solution: Make sure that all for loops in generate statements start from 0.

When a module is instantiated both within and outside of a generate statement, only the breakpoints in the module outside the generate statement are instrumented.

When starting the Identify instrumentor, if you specify a project file (.bsp) as a command line argument without the -f option, any subsequent argument on the command line is ignored.

Solution: Either use the -f flag to specify the project file or always be sure to include the project file name as the last argument on the command line.